

MICROHOBBY

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO II N.º 28

160 Ptas.

Canarias 165 pts.

LOS INCREIBLES
GRAFICOS DEL
AMSTRAD
PASO A PASO

**HOJAS
DE CALCULO:
LA RAZON
DE SER DEL
ORDENADOR
ES
MASTERCALC**

TURBOGRAF:
Representa
tus funciones
en pocos segundos

SOFTWARE

«Lucha en la
batalla decisiva
de la II Guerra
Mundial con
Ratas del desierto»
(Erwin Rommel)



HOBBY PRESS

TU PROGRAMA DE RADIO

claro!



ALDISX2

- Entrevistas a fondo
- Exitos en Soft
- Noticias en Hard
- Concursos

Programátelo: Sábados tarde de 5 a 7 horas.
En directo y con tu participación.

LA COPE A TOPE.

— RADIO POPULAR 54 EMISORAS O.M. —

En Barcelona Radio Miramar



AMSTRAD

sumario

Director Editorial
José I. Gómez-Centurió

Director Ejecutivo
José M.ª Díaz

Subdirector
Juan José Martínez

Redactora Jefe
María García

Diseño gráfico
José Flores

Colaboradores

Javier Barceló

David Sapuerta

Robert Chatwin

Eduardo Ruiz

Francisco Portalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalio Gómez

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Javier Martínez

Portada

Manuel Barco

Ilustradores

J. Igual, J. Pons, F. L. Frontán,

J. Septien, Pejo, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

María Andriño

Consejero Delegado

José I. Gómez-Centurió

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortés

Tel: (93) 303 10 22/313 71 62

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración y Publicidad

La Granja, 39

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Dto. Circulación

Carlos Peropadre

Distribución

Coedis, S. A. Valencia, 245
Barcelona

Imprime

ROTEDEC, S. A. Crta. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solana, 16

Depósito Legal:

M-28468-1985

Derechos exclusivos

de la revista

COMPUTING with

the AMSTRAD

Representante para Argentina, Chile,

Uruguay y Paraguay, Cia.

América de Ediciones, S.R.L. Sud

América 1.532. Tel.: 21 24 64. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidaria de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

Año II • Número 28 • 11 al 17 de Marzo de 1986
160 ptas. (incluido I.V.A.)
Canarias, 155 ptas. + 10 ptas. sobretasa aérea
Ceuta y Melilla, 155 ptas.

5 Primera plana

Libros de aventuras en disco. Nuevos programas de Digital Research para el PCW8256.

6 Primeros pasos

Entre la informática y el arte hay un nexo muy claro, aunque en general poco conocido y explotado. Poca gente sabe que la pantalla de su ordenador es un lienzo lleno de posibilidades de color, y menos gente aún sabe como usarla. Primeros pasos explica cómo.



10 Primeros repastos

Volvemos a explicar qué es un bucle, para qué sirve y cuál es la forma más cómoda de implementarlo: los órdenes WHILE...WEND.

12 ProgramAcción

Es el momento de averiguar cómo trabajan juntas las envolventes de tono y de volumen, y qué sonidos podemos conseguir con ellas.

16 Análisis

GENPANT no es un generador de programas, pero sí un pequeño programa que explica cómo crear pantallas de presentación de datos para nuestros propios programas.

18 Mr Joystick

Ratas del Desierto es un excelente juego de estrategia que nos llevará a la conquista del Norte de Africa tras una dura batalla de inteligencia e ingenio contra nuestro ordenador.



20 Banco de pruebas

Las hojas de cálculo son las herramientas que, por excelencia, hacen imprescindible el uso de un **Amstrad**. Matercalc es uno de los mejores exponentes de este tipo de programas, muy rápido y muy sencillo de usar.



Código Máquina 28

Examinamos en este número las instrucciones que controlan directamente la CPU, el microprocesador Z80 del **Amstrad**.

... Aún más sorpresas
en el n.º 4

ya está en
tu quiosco

AMSTRAD

también disponible
para

SPECTRUM, PLUS, 128

COMMODORE 64

ENDZONE

Infiltrate en el sistema de seguridad del enemigo y destruye todos sus sistemas de comunicación. Cuidado con los radares de control, ya que al ser localizado disparan sobre ti.

AMSSNAKE

Tu amigo Snake está hambriento y se ha puesto en contacto contigo para que le ayudes a encontrar su comida. Procura esquivar las ratas que se crucen en tu camino.

COLOR

Demuestra tu habilidad colocando las bolas del mismo color en las distintas columnas dispuestas para este fin.

DEATH

Conduce a tus amigos a través de caminos y junglas hasta encontrar las cuevas que te salvarán de los peligros. Cuidado con la isla dominada por la Araña Negra.

DOBLEALT

Este programa es una utilidad que te permitirá obtener caracteres de doble altura en cualquiera de los modos de pantalla que posee el Amstrad.

795 pts.
(Incluido IVA)

**YOUR
COMPUTER**

Si no lo encontrara en su quiosco, solicítelo directamente a nuestra editorial.

SMO

Paseo de la Castellana, 268
28046 Madrid. Tel. (91) 733 25 99

La mejor selección de programas de juegos y utilidades, publicados en la revista de mayor difusión de ordenadores de Europa. Ahora reproducidos en cassette, en auténtica exclusiva mundial.



PROGRAMAS DE DIGITAL RESEARCH PARA PCW8256

Digital Research presenta para el 8256 cuatro nuevos programas de gran importancia, dedicados a dos campos de la actividad de este ordenador: los negocios y las utilidades de programación. Al primero pertenecen DR GRAPH y DR DRAW, que, como su nombre indica, sirven para la creación de gráficos de negocios y para dibujar, respectivamente.

En cuanto a las utilidades de programación, Digital Research viene con fuerza, con dos productos casi míticos, un compilador de Pascal, el PASCAL/MT+, y el no menos famoso compilador de Basic CBASIS COMPILER.

Este último no es un compilador para el Basic de **Amstrad**, sino que requiere el uso de su propio dialecto Basic, adaptado más que nada a la gestión, aunque se ha previsto el uso de la extensión GSX del CP/M+ para el tratamiento de gráficos.

LIBROS DE AVENTURAS EN DISCO

Ideologic va a comercializar en España la colección de libros de aventuras TELARIUM... en disco.

La idea viene de América, y está haciendo furor allí.

Se dispone ya de unos cuantos títulos famosos, como «Cita con rama» de Clarke, «Fahrenheit 451», «La isla del tesoro», «El mago de Oz» y otros muchos.

Un libro, enfocado de esta manera, se convierte en algo interactivo, en el cual el «lector» participa a través de pantallas de gráficos y sonido, viviendo la aventura en el más exacto sentido de la palabra.

El libro está arropado por un programa llamado «parser»; es un analizador sintáctico de lenguaje natural, que es capaz de comprender lenguaje escrito y posee un vocabulario de 500 palabras.

Se espera que los primeros títulos aparezcan a partir del mes de mayo.

Primera PLANA

NOVEDADES DE TIMATIC SYSTEMS

La compañía inglesa Timatic Systems ha lanzado discos duros, de los llamados «Winchesters» para los ordenadores **Amstrad** PCW8256, CPC6128 y CPC664, de distintas capacidades medidas en Megabytes.

La disponibilidad de estos periféricos abre posibilidades insospechadas a los **Amstrad** en el campo de su utilización profesional. Sólo nos resta desear que los precios de estos sistemas estén en consonancia con la tónica general de **Amstrad** y que pronto podamos encontrarlos en nuestras tiendas.

Pero no acaba ahí la cosa: también los programadores profesionales recibirán ayuda de Timatic.

Su programador de Eproms ya está casi listo, junto con una tarjeta que puede manipular hasta cuatro ROMs y permite la programación de Basic en ROMs.

COMPILADOR DE BASIC

PSS es una firma de software que, además de hacer juegos, también comercializa utilidades de programación, especialmente compiladores, como la serie de ellos que trabaja en el Spectrum: los Mcoder Compilers.

Pues bien, ahora le ha tocado el turno a **Amstrad**, y ya existe el Mcoder III, un compilador completísimo que acelera muchas veces la velocidad de los programas escritos en Basic, y además admite algo muy importante con lo que sus predecesores no podían: la aritmética de coma flotante, es decir, puede compilar cualquier tipo de número, no sólo los enteros.

Los posibles usuarios, tanto de gestión como de aplicaciones científicas, entenderán enseguida la inmensa ventaja que esto representa.

TRAZOS Y DIBUJOS

Hay quien opina que los ordenadores son unas máquinas frías y calculadoras, muy unidas a la técnica y a la tecnología pero bastante distantes de todo lo relacionado con el arte y la creación. Vamos a intentar demostrar que, usando con un poco de buen gusto nuestro Amstrad, podemos hacer caminar de la mano ambas tendencias: la tecnológica con la artística.



¿Estamos dispuestos a derrochar imaginación? Nuestro ordenador se va a convertir, con un poco de práctica, en un conjunto de paleta, pinceles y colores con el que vamos a ser capaces de realizar verdaderas obras maestras.

Pero antes de nada vamos a establecer las «reglas del juego», o normas de trabajo.

¿Recuerdan la manera que está dividida la pantalla según los conocimientos que tenemos hasta el momento? Ya estamos escuchando, o más bien imaginando su respuesta.

En efecto, en el «Modo» que estamos trabajando habitualmente, la pantalla de nuestro monitor está dividida en 25 filas de 40 columnas. O mejor dicho tenemos 25 líneas y cada una de ellas van a dar cabida a 40 caracteres, uno por columna.

Es decir, hemos partido la pantalla en 1.000 cuadraditos (40×25) iguales en los que podíamos escribir hasta un millar de números, letras o signos de puntuación. Es lo que llamamos «celda de carácter».

Como ya debemos conocer, se puede imprimir cualquier mensaje en un lugar determinado de la pantalla utilizando los números de columna y de línea como coordenadas. Utilizando la instrucción:

LOCATE x, y

colocaremos el cursor de textos —o primera posición para escribir— en la «columna» x y la «fila» y, y a partir de ahí comenzará la escritura.

El origen de este sistema de referencia comenzará en el extremo superior izquierdo de la pantalla. Por tanto, el cuadrado —o celda

de carácter— colocado en ese rincón tendrá como coordenadas 1,1 —fila y columna número 1— y éstas irán aumentando de valor hacia la derecha y hacia abajo. ¿Lo recuerda?

Volvemos a insistirle en que el valor de la coordenada «x» puede tomar valores comprendidos entre 1 y 40 (columna) y el de la coordenada y oscila entre 1 y 25 (fila) siempre que estemos trabajando dentro del modo habitual Mode 1.

En la figura 1 se encontrará una representación gráfica de este sistema de coordenadas así como la manera de determinar exactamente las de una celda de carácter.

La pantalla de alta resolución

Pero una de las características más importante de nuestro Amstrad es la posibilidad que tiene de realizar gráficos con alta resolución. Esto significa que el ordenador va a ser capaz de dibujar rectas, curvas, círculos y de rellenar superficies con la única condición que se lo digamos correctamente.

Nuestro «amigo informático» está dotado de una serie de comandos que nos permiten trazar con gran precisión cualquier línea dentro de la pantalla.

Pero si solamente tuviéramos la precisión que nos dan las 40 columnas por 25 líneas estaríamos muy limitados. Algo habrá que nos haga ser más exactos.

Para los gráficos la pantalla está dividida de distinto modo a como lo estaba para localizar un texto.

Ahora la tenemos dividida en 640 puntos en horizontal y 400 en vertical. En realidad hemos dividido cada cuadrado «de los de antes», o celda de carácter, en 16 filas y 16 columnas. De ahí el número de puntos: $40 \times 16 = 640$ en horizontal y $25 \times 16 = 400$ en vertical.

Podemos definir también ahora un sistema





de coordenadas para los gráficos. Esta vez tomaremos como referencia el extremo inferior izquierdo de la pantalla, que será el punto de coordenadas 0,0. El eje «x» será el horizontal y el «y» el vertical.

Los valores de «x» irán creciendo hasta 639, que nos marca el límite de la pantalla por la derecha. De un modo análogo, los de «y» comenzarían en 0 y subirían hasta 399 que correspondería al borde superior.

Por tanto todo punto de la pantalla se corresponde con una pareja de coordenadas x e y.

0,399

serían las coordenadas de la esquina superior izquierda, mientras que:

639,0

corresponderían a las de la inferior derecha y:

320,200

nos marcarían justamente el centro de la pantalla.

Es bastante evidente que si disponemos de más puntos donde podamos colocar el cursor de los gráficos, paralelamente obtendremos una mayor precisión a la hora de imprimirlos. Solamente como dato anecdótico, si calculamos el número de puntos que hay en la pantalla:

$400 * 640 = 256.000$ puntos

comprenderemos fácilmente el por qué hablar ahora de alta resolución: cualquier línea que dibujemos será tan ancha como un punto.

Por medio de la figura II comprenderá fácilmente lo que le hemos dicho.

Pixels y su tratamiento: comando Plot

A cada uno de estos puntos en los que hemos dividido la pantalla lo vamos a llamar a partir de ahora «pixel» y es la mínima superficie cuyo color o posición podemos controlar individualmente: es el espacio reservado para colocar el gráfico más pequeño que existe un punto.

Bueno, ya conocemos la manera de fijar las coordenadas de un punto dentro de la pantalla. Vamos a ver la forma de poder «iluminarle» o «apagarle», es decir, dibujarle o trazarle.

El **Amstrad** tiene, como siempre, una herramienta que nos va a permitir hacerlo sin ningún problema. Se trata de la instrucción **PLOT**. Su forma más sencilla es:

PLOT x, y

donde x e y son las coordenadas gráficas de la posición de la pantalla donde queremos colocar el punto.

PLOT «ilumina» el «pixel» colocado en la columna «x» y en la línea «y» dentro de esta pantalla gráfica.

Vamos a detenernos un momento. Observe un detalle: no es lo mismo la pareja de nú-

Primeros PASOS

meros que siguen al comando **PLOT** que la pareja que acompaña a **LOCATE**.

Las coordenadas de **PLOT** tienen como origen la esquina inferior izquierda y van en orden creciente hacia la derecha y hacia arriba. Las de **LOCATE** comienzan en la esquina superior izquierda y aumentan hacia la derecha y hacia abajo.

Otra diferencia radica en el rango de sus valores. Mientras que **PLOT** llega a 639 hacia la derecha y a 399 hacia arriba, siempre dentro de la pantalla, **LOCATE** utiliza hasta 40 a la derecha y 25 hacia abajo. **¿Ven la diferencia?** Continuemos:

PLOT x, y

marcará en color una porción muy pequeña de la pantalla que está localizada en la posición de coordenadas x, y. Podemos hacer **PLOT** a cualquier punto. Pero, **¿qué pasará si las coordenadas están fuera de nuestro monitor?**

Pruebe a teclear:

PLOT 700,600

y observe que ocurre en la pantalla. Parece que no ha ocurrido nada pero en realidad, aunque está fuera, hemos colocado el cursor gráfico en un determinado punto (el de coordenadas 700,600) para posteriores instrucciones gráficas. Ya veremos más adelante para qué nos puede servir esto.

Con el programa uno creamos un cielo magníficamente estrellado.

Programa uno

Pero lo que empezó como un bonito paisaje, poco a poco se nos va deformando y convirtiéndose en una gran mancha borrosa. **¿A qué es debido?** Veamos como funciona.

Con las líneas 30 y 40 calculamos aleatoriamente las coordenadas gráficas x e y del pixel que queremos iluminar y a continuación —línea 50— lo marcamos en la pantalla.

Y así indefinidamente. Debido a este bucle sin salida, con un poco de paciencia llegaríamos a ver toda la pantalla coloreada. No serían estrellas, veríamos un sarpuellido peligroso que nos invade el monitor. ¡Es broma!

¿Y para iluminar, o trazar, una recta? En principio vamos a pensar que una recta es una sucesión de puntos. Pues manos a la obra, nos bastará con **PLOT**ear todos y cada uno de los puntos de la línea.

En realidad, obtendríamos una serie discontinua de puntos pero la sensación que recibe

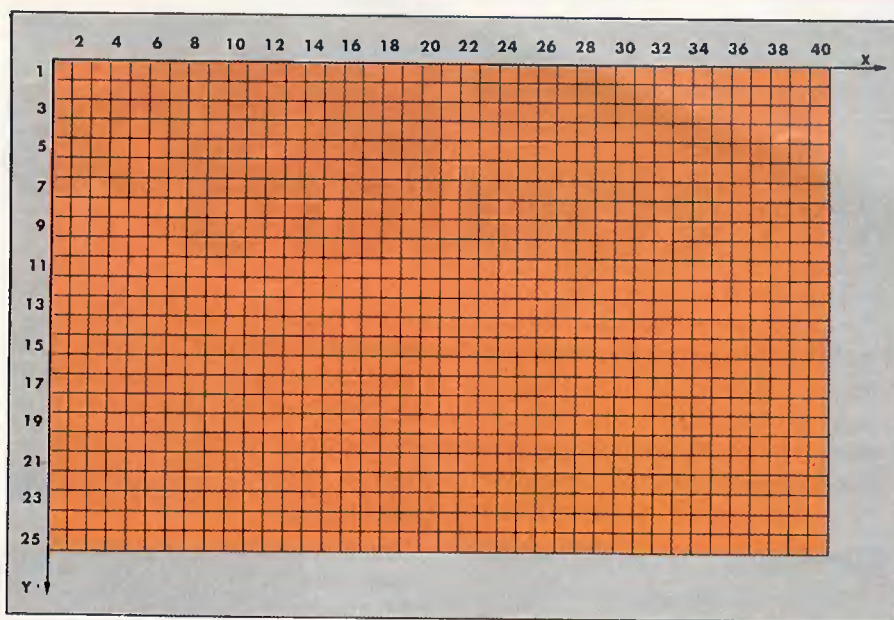


Fig. I: Pantalla de texto; Modo 1 (40 columnas).

nuestra vista es la de tener en la pantalla una línea recta. Con el programa dos podemos comprobarlo.

Programa dos

En la primera parte del programa trazamos la línea punto a punto con una instrucción PLOT para iluminar cada pixel. Basta con dar las coordenadas del contiguo y los puntos se van uniendo y dando forma a la recta líneas 50 a 160. Evidentemente ésta es una manera muy rupestre de trazarla ya que necesitamos una instrucción por cada punto y se pueden imaginar el «programon» que resultaría en cuanto la línea sea un poco larga. De ahí el ETC. que sitúa a la sucesión de puntos producido por la línea 180.

Si nos ha seguido en anteriores artículos con un poco de interés, comprenderá que esta serie de instrucciones PLOT podemos convertirla en una bucle semejante al que forma la segunda parte del programa líneas 210 a 230.

Utilizamos «bucle» como variable de control a la vez que va dando valores a las coordenadas de los puntos contiguos que dan forma a la recta línea 220.

Intente variar el límite superior del bucle —línea 210— dándole cualquier otro valor. Verá como cambia la longitud que alcanza la serie de pixels iluminados.

Observe que utilizamos un pequeño truco para que no nos aparezca el mensaje READY y nos descoloque la figura en la pantalla. Por medio del bucle de la línea 240, que no tiene salida, nos metemos en un circuito repetitivo que no hace nada. El valor de «variable» no cambia y se mantiene siempre a 0 y, por tanto al aplicarle el operador lógico NOT nos da verdadero continuamente y el bucle se repite sin fin. Ingenioso, ¿no?

No olvide que estas aparentes rectas son en realidad una sucesión de puntos coloreados

y por tanto no forman un trazo continuo. Si pasa por alto este detalle quizá se encuentre, sin esperarlo, una sorpresa desagradable.

Para darnos una idea de la situación de un determinado pixel sobre la pantalla vamos a utilizar el programa tres.

Programa tres

Trazamos los ejes de coordenadas por medio de un bucle como en el programa anterior líneas 30 a 80.

A continuación se nos piden los valores de «x» e «y» por medio de los INPUT de las líneas 180 y 200 y con estas iluminamos el punto correspondiente en la 210. Repetimos el proceso para ver la colocación de diversas coordenadas.

Si le parece puede estudiar la manera como borramos los mensajes donde se nos indi-

ca el dato que nos pide el ordenador. Simplemente con un PRINT de espacios en blanco en las líneas 140 y 160 solucionamos el problema.

Este modo de trazar líneas es bastante útil a la hora de dibujar curvas pero si se trata de rectas nuestro ordenador posee otra herramienta que nos facilita mucho el trabajo se trata de la instrucción:

DRAW x, y

que nos va permitir dibujar una línea recta desde el punto donde se encuentra el cursor de gráficos hasta el que tiene por coordenadas los valores de «x» e «y».

El origen estará en el punto donde lo hayamos colocado mediante la instrucción PLOT o donde termine la instrucción DRAW anterior.

Mediante el programa cuatro trazamos los bordes de la pantalla por medio de DRAW.

Programa cuatro

Para trazar una línea es necesario primero fijar el punto de partida. Por eso, en este caso, hacemos el PLOT de la línea 30 y decimos al **Amstrad** que el origen de nuestra línea va a estar en el pixel de coordenadas 0,0. Y a continuación dibujamos la línea hasta el punto 639, 0— esquina inferior derecha de la pantalla.

Por medio de la subrutina que empieza en la línea 1.000 hacemos que el programa espere a que pulsemos la tecla RETURN antes de continuar el dibujo.

Pero PLOT no es la única manera de indicarle al ordenador el comienzo de una línea. Después de la primera instrucción DRAW, el cursor de gráficos queda situado en el pixel donde termina la línea anteriormente trazada. Por eso no es necesario decir el origen de la recta a la instrucción DRAW de la línea 60 para que comience donde finalizará la anterior.

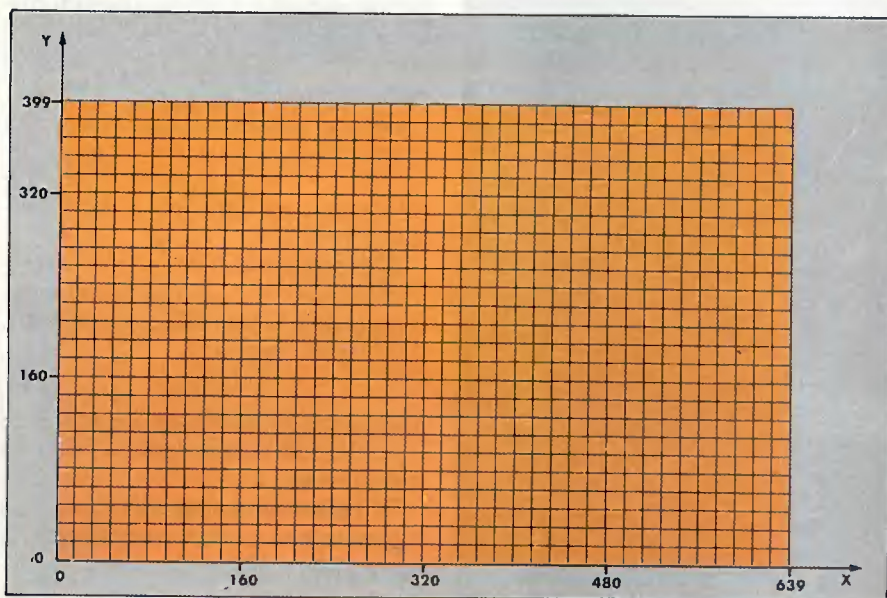


Fig. II: Pantalla de gráficos (640x400 pixels).

Y de un modo análogo trazáramos el resto de las rectas que forman parte de la línea cerrada que bordea la pantalla.

Cuando no hemos dibujado ningún punto, DRAW toma como origen de la recta el punto 0,0. Por esto en este programa se podría suprimir la línea 30 pero quizá por claridad y para facilitar la lectura del mismo sería conveniente dejarlo como está.

El programa cinco es una variante del anterior. La diferencia existente consiste en la manera de darle a DRAW el final de la línea que va a trazar. En esta ocasión lo hacemos mediante un bucle que lee (READ) cada uno de los datos de la línea de DATA-120.

Programa cinco

Observe que en ambos casos utilizamos un bucle sin salida WHILE... WEND, que no hace nada, para conseguir mantener el dibujo intacto en la pantalla.

¿Qué le parece si ahora dibujamos un triángulo equilátero en el centro de la pantalla? Es una tarea que le deja a usted. Pero, como siempre y por si acaso tiene dudas, hemos hecho el programa seis, sólo para que le sirva como guía en el caso que no sepa como «hincarle el diente» a este problema. Lo podrá seguir fácilmente pero, por favor, intente hacer una labor creativa, ya que estamos en plan artístico.

Programa seis

A estas dos instrucciones que hemos comentado —PLOT y DRAW— podemos añadirles un parámetro más. Se trata de darles color. La forma de hacerlo sería:

PLOT x, y, color

o bien

DRAW x, y, color

donde el parámetro «color» ha de ser uno de los cuatro que tenemos disponibles en el Modo que estamos trabajando Modo 1. En este caso podremos dar a Color valores de 0 a 3. Si le damos un superior, el Amstrad lo reduce a uno de estos cuatro. Recuerde que todo esto corresponde a Modo 1.

El programa siete es una demostración de cómo darle color a sus gráficos. Pruebe.

Programa siete

Vea en la línea 70 la manera que ponemos el parámetro de color y cómo le vamos dando valores por medio de la variable de control del bucle FOR... NEXT de las líneas 30 y 90.

Nuestro «cuadro abstracto» consiste en una serie de líneas rectas cuyo punto final es



aleatorio dentro de la pantalla. Elegimos las coordenadas al azar —líneas 50 y 60. Repetimos el proceso tantas veces como nos indiquen los límites de la variable de control del bucle FOR... NEXT de las líneas 40 y 80.

Y esta operación la repetimos para cada uno de los tres colores distintos del fondo según el bucle de las líneas 30 y 90. ¿Le queda bonito?

Y ya nos vamos a despedir. **¿Conoce los diagramas de barras?** Seguro que sí. Le vamos a pedir un favor: ¿le importa comentarnos la forma cómo los hace el programa ocho? **¿Verdad que es una herramienta muy útil?**

La próxima semana seguiremos.

PROGRAMAS

```
10 REM PROGRAMA I
20 CLS
30 x=INT(RND*640)+1
40 y=INT(RND*400)+1
50 PLOT x,y
60 GOTO 30
```

```
10 REM PROGRAMA II
20 CLS
30 LOCATE 1,23
40 PRINT"LINEA HECHA PUNTO A PUNTO"
50 PLOT 0,0
60 PLOT 1,0
70 PLOT 2,0
80 PLOT 3,0
90 PLOT 4,0
100 PLOT 5,0
110 PLOT 6,0
120 PLOT 7,0
130 PLOT 8,0
140 PLOT 9,0
150 PLOT 10,0
160 PLOT 11,0
170 LOCATE 15,25
180 PRINT"ETC....."
190 LOCATE 1,11
200 PRINT"LINEA HECHA MEDIANTE UN BUCLE"
210 FOR bucle=0 TO 639
220 PLOT bucle,200
230 NEXT bucle
240 WHILE NOT variable:WEND
```

```
10 REM PROGRAMA III
20 CLS
30 FOR x=0 TO 640
40 PLOT x,0
50 NEXT x
60 FOR y=0 TO 400
70 PLOT 0,y
80 NEXT y
90 LOCATE 40,24
100 PRINT"x"
110 LOCATE 2,1
120 PRINT"y"
130 LOCATE 10,1
140 PRINT"
```

```
150 LOCATE 10,2
160 PRINT"
```

```
170 LOCATE 10,1
180 INPUT"COORDENADA X DEL PUNTO: "
,x
190 LOCATE 10,2
200 INPUT"COORDENADA Y DEL PUNTO: "
,y
210 PLOT x,y
220 GOTO 130
```

```
10 REM PROGRAMA IV
20 CLS
30 PLOT 0,0
40 DRAW 639,0
50 GOSUB 1000:REM CONTINUAR
60 DRAW 639,399
70 GOSUB 1000:REM CONTINUAR
80 DRAW 0,399
90 GOSUB 1000:REM CONTINUAR
100 DRAW 0,0
110 LOCATE 15,12
120 PRINT" OK "
130 WHILE NOT variable:WEND
140 END
1000 REM CONTINUAR
1010 LOCATE 15,12
1020 INPUT"FULSA RETURN ",tecla$
1030 RETURN
```

```
10 REM PROGRAMA V
20 PLOT 0,0
30 CLS
40 FOR bucle=1 TO 4
50 READ x,y
60 DRAW x,y
70 GOSUB 130:REM CONTINUAR
80 NEXT bucle
90 LOCATE 15,12
100 PRINT" OK "
110 WHILE NOT variable:WEND
120 DATA 639,0,639,399,0,399,0,0
130 REM CONTINUAR
140 LOCATE 15,12
150 INPUT"FULSA RETURN ",tecla$
160 RETURN
```

```
10 REM PROGRAMA VI
20 CLS
30 PLOT 200,95
40 DRAW 440,95
50 DRAW 320,95+INT(240*SQR(3)/2)
60 DRAW 200,95
```

```
10 REM PROGRAMA VII
20 CLS
30 FOR tinta=1 TO 3
40 FOR linea=1 TO 30
50 x=INT(RND*640)
60 y=INT(RND*400)
70 DRAW x,y,tinta
80 NEXT linea
90 NEXT tinta
```

```
10 REM PROGRAMA VIII
20 CLS
25 INPUT"ALTURA DE LA FRANJA: ",altura
ura
30 FOR linea=0 TO altura
40 PLOT 280,linea
50 DRAW 360,linea
60 NEXT
```


Los BUCLES

Ni un solo programa funciona sin incorporar en su interior tareas repetitivas, bucles. Una de las mejores formas de hacerlo es usar los bucles estructurados, como el WHILE.. WEND.



Vamos a introducirnos ahora en algo completamente distinto. Sin duda se recordarán programas del tipo del programa 1, es decir, bucles sin fin o bucles incondicionales.

Hay por lo menos otra manera de obtener bucles incondicionales, la del programa 2.

Al ejecutarlo, la palabra «hola» será impresa repetidamente hasta que se pulse la tecla ESC.

Los comandos WHILE marcan el comienzo y el final de un sector de programa que debe ser repetido.

A este sector le llamaremos el Cuerpo del bucle.

Hay que colocar una expresión detrás del comando WHILE, que el **Amstrad** evaluará para saber si es cierta o falsa, esto es, si es 1 ó 0. La sección de programa marcada con WHILE... WEND se ejecutará mientras la expresión sea cierta. Cuando sea falsa, el flujo del programa pasa a la sentencia inmediatamente siguiente a WEND.

La línea 30 del programa 6 dice:

```
30 WHILE 1=1
```

Literalmente le estamos diciendo al **Amstrad**: «hazlo mientras uno sea igual a uno».

El programa 3 es virtualmente idéntico al 6, excepto por la línea 30, que ahora dice:

```
30 WHILE -1
```

Al correrlo se obtendrá exactamente el mismo resultado, sorprendentemente.

Algo menos de sorpresa experimentaremos si tecleamos:

```
PRINT 1=1
```

Aparecerá -1.

La razón es que a las computadoras les encantan los números, y el **Amstrad** asigna al concepto verdad o cierto el valor -1. Por tanto, en ambos programas la línea 30 significa «mientras verdad».

Por otro lado, al teclear:

```
PRINT 1=2
```

se verá un rotundo 0 en la pantalla.

Al cambiar la línea 30 por:

```
30 WHILE 0
```

queremos repetir un trozo de programa Basic mientras (lo que sea), sea falso. Por supuesto, el bucle no se ejecuta ni una vez.

Bucles condicionales

El programa número 4 usa un bucle WHILE... WEND que evalúa una condición muy simple: las líneas 40 y 50 del programa serán repetidas hasta que la variable fin\$ sea igual a «sí», o, lo que es lo mismo, hablando en términos del propio bucle, mientras la variable fin\$ sea distinta de «sí», se repite el proceso.

Aunque esta variable sea mentada por primera vez en la línea 30, el **Amstrad**, la hace igual a cadena vacía, esto es, a nada, automáticamente. Por eso no necesitamos definirla previamente y, además, sabemos que en la primera iteración del bucle la condición especificada no se cumplirá; al menos se ejecutará una vez.

Exactamente lo mismo habría ocurrido con una variable numérica. Atención ahora al programa 5. En él, iniciamos la variable número a 0 (línea 30) e imprimimos su valor (línea 40).

Entonces le sumamos 1 y atesoramos el resultado en la nueva variable «nuevo número», imprimiéndolo (líneas 50-60).

Con la línea 50 del 5 en la mente, examinemos el número 6. El trabajo que realiza es prácticamente el mismo que el programa anterior y su lógica está basada en él. Hemos hecho, sin embargo, un cambio importante, también en la línea 50.

Vamos a analizarla un poco, para tratar de comprender, entre otras cosas, lo que significa una igualdad para un ordenador:

```
50 número=número+1
```

Desde un punto de vista estrictamente matemático, esta sentencia no tiene sentido. No se puede decir que algo es igual a sí mismo más uno, como si dijéramos, por ejemplo, $5=5+1$.

Primeros repastos

```
10 REM programa 1      10 REM programa 2
20 MODE 1              20 MODE 1
30 PRINT "hola"         30 WHILE 1=1
40 GOTO 30              40 PRINT "hola"
                        50 WEND
```

```
10 REM programa 3
20 MODE 1
30 WHILE -1
40 PRINT "hola"
50 WEND
```

```
10 REM programa 4
20 MODE 1
30 WHILE fin$<>"sí"
40 PRINT "Esto es un bucle WHILE...
WEND condicional"
50 INPUT "Quieres acabar?";fin$
60 WEND
70 PRINT "Ya me voy"
```

```
10 REM programa 5
20 MODE 1
30 número=0
40 PRINT "Número es ";número
50 nuevonúmero=número+1
60 PRINT "El nuevo número es ahora "
;número
```

```
10 REM programa 6
20 MODE 1
30 número=0
40 PRINT "Número es ";número
50 número=número+1
60 PRINT "El nuevo número es ahora "
;número
```

Sin embargo, aquí se esconden dos importantes claves:

1. El micro evalúa en primer lugar la parte derecha de la expresión, esto es, «número+1». La primera vez que la máquina llega a la línea 50, mira el valor de número, que es cero y le suma 1. Tenemos pues, que la parte derecha vale uno.

2. Para el **Amstrad**, el signo «=», significa asignación en lugar de igualdad, normalmente. Toma, por tanto, el valor de la parte derecha, 1, y lo asigna a la variable número. Como el valor anterior de número era cero, ahora será 1. Si aquí hubiera un bucle, se repetiría el cálculo, la segunda pasada asignaría a número el valor 2, resultado de tomar el valor actual (1) y sumarle 1.

En definitiva, la línea 50 aumenta el valor de la variable número en una unidad.

El ordenador no tiene ningún problema en tomar el valor de una etiqueta (variable), usarlo para un cálculo y cambiar el valor de esa misma etiqueta. Por eso a las variables se las llama, precisamente, variables.

Te ofrecemos algo "muy Especial"

En el mes de septiembre nació una **AMSTRELLA** que vino a demostrar que los **Amstrad** estaban ahí y había que contar con ellos.

Hoy, miles de personas nos dan la razón.

Por este motivo, y después de los 6 primeros meses de andadura juntos con nuestros lectores, ha vuelto a suceder algo muy importante: ha nacido una **AMSTRELLA MUY ESPECIAL**.

¡Ya está a la venta en tu quiosco!
por sólo **350** ptas.



HOBBY PRESS
Para gente inquieta.

¿QUE SE PUEDE HACER CON UN ORDENADOR?

Rellena este cupón y envíalo a **HOBBY PRESS, S. A.** Apartado de Correos 232. Alcobendas. Madrid.

Nombre _____ Apellidos _____ Profesión _____ Provincia _____

Domicilio _____

Localidad _____

Teléfono _____

¿Eres suscriptor de **MICROHOBBY AMSTRAD**? ☐ Si ☐ No ☐

DESEO RECIBIR

FORMA DE PAGO

☐ Talón bancario adjunto a nombre de **HOBBY PRESS, S. A.**

☐ Mediante Tarjeta de Crédito. N.º _____

☐ Contra reembolso (supone 75 ptas. de gastos de envío).

Fecha y firma: _____

Edad _____ C. Postal _____

Envío a precio de 350 ptas (IVA incluido)

LOS SECRETOS DE LAS ENVOLVENTES DE TONO Y VOLUMEN

Esta semana avanzaremos un poco más, veremos con mayor profundidad las envolventes de volumen y tono que exploramos en los dos últimos artículos.



Vamos a reencontrarnos con alguno de nuestros viejos amigos viendo a la vez cómo trabajan. Tecleemos:

ENV 1,5,2,20

para definir la envolvente de volumen uno. Estará formada por cinco escalones, cada uno de los cuales durará un quinto de segundo (20/100). Al principio de cada escalón el volumen se incrementará en dos.

Seguidamente crearemos una envolvente de tono con:

ENT 1,5,10,20

También estará compuesta por cinco escalones y cada uno de ellos durará un quinto de segundo. Cuando comienza cada escalón, el parámetro de tono se incrementa en 10 unidades.

Ahora vamos a escuchar ambas en acción con:

SOUND 1,200,100,5,1,1

Produce un sonido en el canal A durante un segundo. Recuerde que los comandos SOUND tienen un tono inicial de 200 y un volumen de 5.

Ambos valores se ven afectados por los dos parámetros finales del comando SOUND que hacen posible que la envolvente de volumen número 1 y la envolvente de tono número 1 ejerzan sus influencias sobre el sonido producido.

El resultado es una nota que dura un segundo cuyo tono decrece mientras que su volumen se incrementa.

Ahora vamos a deshacernos de la envolvente de volumen 1 introduciendo:

ENV 1

y pulsando ENTER. Definir una envolvente de volumen sin parámetros es la forma más eficaz de suprimirla.

Si probamos de nuevo:

SOUND 1,200,100,5,1,1

seguiremos escuchando una nota cuyo tono va bajando cinco escalones pero el volumen se mantendrá constante. La envolvente de tono sigue trabajando pero los efectos producidos por la envolvente de volumen han desaparecido.

Del mismo modo: ENT 1

hace desaparecer la envolvente de tono número 1.

Repetimos: SOUND 1,200,100,5,1,1

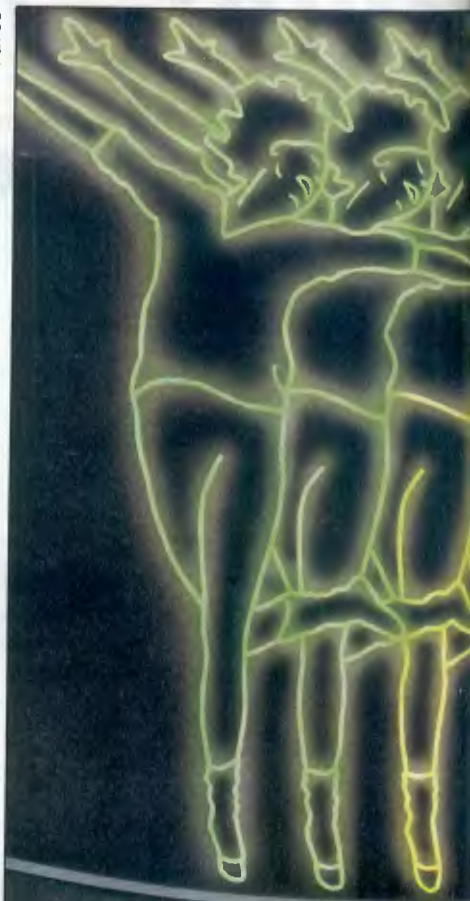
y obtendremos una nota que suena durante un segundo con un tono constante de 200 y un nivel de volumen igual a 5. Los parámetros de las envolventes no producen ningún efecto ya que hemos destruido las nuestras.

Vamos a ver de cerca una envolvente de volumen. Redefinamos la número 1 con:

ENV 1,5,2,10

Como tiene cinco escalones y cada uno de ellos dura una décima de segundo, la duración total de la envolvente será de medio segundo.

M. Barco





Escuchemos sus efectos en:

SOUND 1,200,50,5,1

¿Recuerda qué ocurre si el parámetro que indica la duración del comando SOUND excede la duración total de la envolvente de volumen?

O sea, **¿si SOUND dura más que la envolvente?**

Podemos averiguarlo con:

SOUND 1,200,100,5,1

que dura un segundo entero.

Como podemos escuchar, la envolvente de volumen funciona igual que antes durante el primer medio segundo. A partir de aquí el efecto de la envolvente se detiene y el sonido continúa durante el medio segundo restante con un nivel de volumen constante.

En otras palabras, la envolvente sólo trabaja una vez. No se repite dentro del mismo comando SOUND.

Si queremos podemos hacer que sí se repita. Es fácil y conseguiremos producir algunos efectos interesantes.

Pero antes vamos a admitir otra vez que no hemos sido del todo honrados con usted.

Si puede, vuelva al artículo donde le decíamos que el parámetro de duración del comando SOUND puede tener un rango desde 0 a 32767.

Esto no es así. En realidad dicho parámetro podrá valer entre -32768 y 32767. Queremos decir con esto que puede ser un número negativo.

A primera vista puede parecernos una cosa bastante absurda. Las duraciones de 0 a 32767 están de acuerdo con el sentido común, pero **¿qué hay de los valores negativos? ¿Producen sonidos con «marcha atrás»?**

Lo que ocurre en realidad es que no consideramos los valores negativos como la cantidad de tiempo que suena la nota sino como el número de veces que se va a repetir la envolvente de volumen.

El signo negativo le dice al **Amstrad** que el número que le sigue no es una duración sino un contador que indica el número de veces que se cumple dicha envolvente.

Si el parámetro de duración es -3 entonces se repite 3 veces. Si es -100 quiere decir que la envolvente de volumen actuará 100 veces sobre el sonido.

No se limite a creer nuestras palabras y punto. Compruébelo.

Introduzca: ENV 1,5,2,10

y

SOUND 1,200,-2,5,1

y escúchelo usted mismo. El -2 en el parámetro de duración obliga a que se repita dos veces la envolvente de volumen (invocada por el último parámetro del comando SOUND).

Puesto que en este caso la envolvente dura medio segundo, el sonido se escuchará un segundo completo.

SONIDO

Del mismo modo:

SOUND 1,200,-1,5,1

durará medio segundo (una repetición de la envolvente dura exactamente este tiempo), y

SOUND 1,200,-20,5,1

durará 10 segundos.

Como podemos ver y escuchar, cuando el parámetro es negativo la duración del sonido se determina por la de la envolvente de volumen y el número de veces que se repite. Por lo tanto podemos controlar la longitud del sonido mediante la envolvente.

Observe, sin embargo, que al comenzar a repetirse, el sonido no regresa al nivel inicial de volumen sino que continúa en el nivel con el que terminó. Cuando el valor del parámetro de volumen se sale fuera de rango, hace como las agujas del reloj, vuelve a empezar en el otro extremo del rango.

Pongamos un ejemplo de la envolvente de volumen: ENV 1,5,2,20

y

SOUND 1,200,-2,5,1

En esta ocasión la envolvente actuará dos veces sobre el sonido. Cabría esperar que los volúmenes valgan 7, 9, 11, 13 y 15 en la primera repetición y 7, 9, 11, 13 y 15 para la segunda. Pero no es esto lo que ocurre.

La primera aplicación de la envolvente produce niveles de volumen con valores 7, 9, 11, 13 y 15 pero la segunda pasa por 0 y continúa produciendo los valores 1, 3, 5, 7 y 9. Vea que el parámetro de volumen gira como las agujas del reloj —que cuando llega a las 12 se convierte en 0 y vuelve a empezar— dentro de los valores externos del rango.

Si encuentra alguna dificultad en seguirlo, pruebe: ENV 1,5,2,100

y

SOUND 1,200,-2,5,1

y podrá escuchar lo que ocurre exactamente.

La envolvente de volumen afecta al comando SOUND del mismo modo en cada una de las repeticiones, pero los volúmenes resultantes son diferentes porque empiezan a trabajar con un nivel de volumen distinto. Esto hace que el efecto de repetir una envolvente sea difícil de pronosticar.

Sin embargo, al definir una envolvente de volumen no tenemos por qué preocuparnos de repetirla un número exacto de veces sino preocuparnos en calcular el parámetro de duración de una nota. Sonará tanto tiempo como nos indique la envolvente de volumen y el número de veces que la repitamos y no más.

Esta condición puede endulzarnos la vida, especialmente si cambiamos la duración de cualquier escalón de la envolvente de volumen. La simple repetición modifica automáticamente la longitud del sonido de forma que está en línea con el cambio efectuado.

Antes de dejar las envolventes de volumen probemos:

SOUND 1,200,—3,5

y

SOUND 1,200,—10,5

Aparentemente le estamos diciendo al **Amstrad** que repita una envolvente de volumen pero sin especificar cuál es. **¿Sabe qué va a pasar?**

La respuesta es que estamos utilizando una envolvente de volumen. Se trata, por defecto, de la envolvente 0 y hace que la nota suene durante dos segundos al volumen especificado en el comando SOUND.

Mientras que el primer sonido se escucha durante seis segundos —tres repeticiones— el segundo lo oímos durante 20 —10 repeticiones.

Y ahora, que estamos hablando de la repetición de las envolventes, sospechará que continuaremos haciéndolo sobre la repetición de las de tono. Y así es.

Definamos una envolvente de tono con:

ENT 1,5,10,20

y oigamos sus efectos con:

SOUND 1,200,100,5,0,1

El tono del sonido baja cinco escalones. Cada escalón duran un quinto de segundo, así que la envolvente de tono dura un segundo, que es justo el tiempo necesario para reproducir la nota completa. **¿Qué ocurre si la envolvente de tono no dura tanto como el comando SOUND?**

Como siempre, si lo probamos lo veremos en la práctica. En esta ocasión usamos:

ENT 2,5,10,10

y

SOUND 1,200,100,5,0,2

Aquí la envolvente dura medio segundo mientras que el sonido dura un segundo entero.

Como escuchamos, las envolvente actúa durante el primer medio segundo, mientras que el sonido continúa con el tono final constante hasta que se agota el tiempo de duración.

En otras palabras, la envolvente de volumen se utiliza una sola vez y por tanto el sonido continúa sin gozar del beneficio de su ayuda durante medio segundo más.

Podemos, sin embargo, hacer que la envolvente de tono se repita. Vamos a intentar definir una con: ENT —2,5,10,10

Le parecerá poco usual, pero el **Amstrad** la aceptará, con el signo menos y todo. Este signo menos dice al **Amstrad** que cuando está envolvente de tono sea llamada por algún comando SOUND, si es posible, se repita.

Pruébelo con:

SOUND 1,200,100,50,2

Ahora la envolvente de tono se repite todo el tiempo que suena la nota. Puesto que la envolvente de tono dura medio segundo y la duración del sonido es de un segundo, la envolvente se repite dos veces.

Observe que llamamos a la envolvente justamente con la cifra 2 en el comando SOUND, sin hacer uso del signo menos. Intente:

ENT —3,10,5,100

con SOUND 1,100,1000,5,0,3

y ENT —4,5,20,20

con SOUND 1,200,150,5,0,4

Como podemos oír, la envolvente de tono se repite tantas veces como pueda dentro de la duración del comando SOUND.

Fijese que la repetición de envolventes de tono es muy diferente a la repetición de las de volumen.

Con envolvente de volumen la longitud de las mismas y el número de veces que se repiten es lo que determina la duración del sonido producido.

En el caso de las envolventes de tono, el parámetro de duración del comando SOUND mantiene su influencia. La envolvente de tono del tiempo indicado por dicho parámetro. Cuando se pasa del tiempo, la envolvente deja de repetirse, se para.

Ambas envolventes tienen algo en común, cuando se repiten no regresan al principio, sino que continúan desde donde se quedaron.

Tomemos el caso de:

ENT —1,5,10,20

y

SOUND 1,200,200,5,0,1

Aquí la envolvente de tono se repite dos veces.

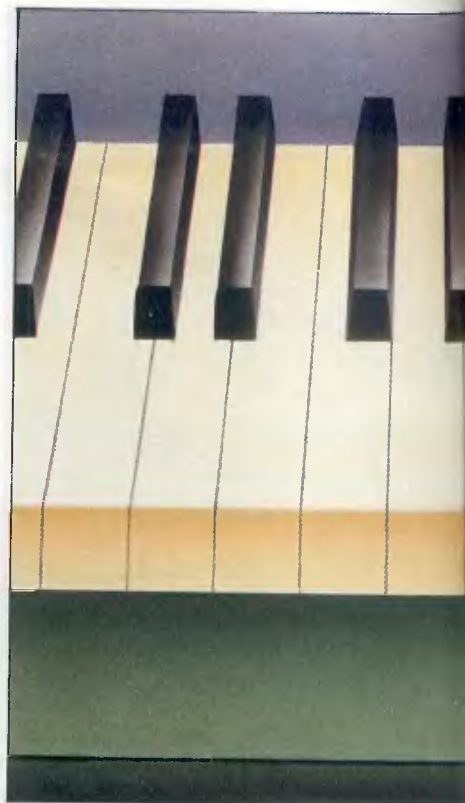
De nuevo podemos pensar que el tono del sonido tomaría los valores 210, 220, 230, 340 y 250 la primera vez, y 210, 220, 230, 240, 250 en la segunda repetición. Y otra vez, ¡nos equivocamos!

Cuando la envolvente de tono comienza por segunda vez no regresa al valor del parámetro de tono del comando SOUND, sino que continúa desde donde se quedó.

Por lo tanto los valores de tono producidos en 210, 220, 230, 240 y 250 la primera vez que se recorre la envolvente y 260, 270, 280, 290 y 300 la segunda.

Hemos obtenido una serie de 10 notas diferentes en lugar de las dos idénticas de cinco que erróneamente estábamos esperando.

Esta circunstancia, especialmente cuando se combina con los efectos del «recorrido» de las agujas del reloj —paso por cero— en los extremos del rango del parámetro de tono puede hacerse muy difícil el pronóstico de lo que va a ocurrir cuando se repite la envolvente.



Hasta aquí hemos dado la definición de las envolventes de tono en la forma:

ENT S, T, V, W

Hay otra manera en la que, en lugar de utilizar escalones para obtener las notas que queremos, solamente le decimos al **Amstrad** qué nota es la que va a sonar.

Tiene la forma:

ENT=tono, duración

donde la fórmula **tono, duración** puede repetirse hasta cinco veces.

Aquí **tono** es el tono real que queremos que tenga la nota. Toma los valores comprendidos dentro del mismo rango que los del parámetro del comando SOUND y trabaja también de la misma manera.

La **duración** oscila entre 0 y 255, diciéndole al micro cuánto ha de durar la nota.

Como sabemos, definiendo una envolvente de tono en la forma tradicional,

ENT 1,5,10,20

y llamándola con:

SOUND 1,200,100,5,0,1

obtenemos un sonido compuesto por cinco notas cuyos respectivos tonos son 210, 220, 230, 240 y 250.

Podemos conseguir el mismo efecto definiendo la envolvente de tono de la manera alternativa que hemos visto partiendo de todos estos valores.

ENT

2, = 210,20, = 220,20, = 230,20, = 240,20, = 250,20



y

SOUND 1,200,100,5,0,2

produce exactamente el mismo sonido. En realidad, cuando definimos la envolvente de tono de este modo, se ignora el valor del parámetro de tono del comando SOUND. Veamos que:

SOUND 1,0,100,5,0,2

produce exactamente las mismas notas. A pesar de la forma alternativa de la definición, la envolvente 2 se comporta de un modo muy parecido a la envolvente de tono ordinaria.

SOUND 1,200,50,5,0,2

terminaría antes de recorrer por completo la envolvente de tono, su duración es menor.

Sin embargo:

SOUND 1,200,400,5,0,2

estaría recorriéndola durante un segundo. Después el sonido se mantendría con el tono final durante tres segundos.

Donde si existe diferencia cuando usamos esta definición alternativa es a la hora de utilizarla de manera que la envolvente se repita.

Definamos una envolvente de tono con repetición:

ENT

-2, = 210,20, = 220,20, = 230,20, = 240,20, = 250,20

A continuación vemos los efectos con:

SOUND 1,200,400,5,0,2

Como seguramente esperábamos, la envolvente se repetirá cuatro veces ya que dura un

segundo mientras el comando SOUND tiene el parámetro de duración igual a 4.

Lo que podíamos no haber previsto es que cada una de las veces que se repite, los diferentes tonos de las notas son los mismos. Van 210, 220, 230, 240, 250, seguidos de 210, 220, 230, 240, 250 y así sucesivamente.

Esta nueva forma de definir la envolvente de tono no utiliza escalones, pero está relacionada con el tono absoluto de las notas.

En cada repetición de nuevo volvemos al principio y va siguiendo la misma secuencia que había hecho anteriormente repitiendo cada una de las notas con sus mismos tonos. De esta manera es mucho más fácil predecir qué hará la envolvente.

Pero vamos a complicar la cosa intentando adivinar qué ocurre cuando repetimos envolventes de tono y de volumen al mismo tiempo. **¿Podríamos pronosticar qué efectos producirán las siguientes instrucciones?**

ENV 1,5,2,20

ENT -1,5,10,20

SOUND 1,200, -4,5,1,1

Permítanos que lo dudemos. Sin embargo, si utilizamos la forma alternativa de definir la envolvente de tono, las cosas serían más fáciles:

ENV 1,5,2,10

ENT

-2, = 210,20, = 220,20, = 230,20, = 240,20, = 250,20

SOUND 1,200, -4,5,1,2

A pesar de esta definición puede seguir resultando complicado el hacer un pronóstico acertado. Pruébalo.

Y es aquí donde dejamos las envolventes de tono y de volumen. Si nos permite vamos a proponerle un pequeño juego.

Hasta ahora definíamos nuestras envolventes y no las tocábamos ni variábamos mientras el sonido se reproducía. **¿Qué ocurriría si las redefiniéramos mientras están sonando? ¿Por qué no lo probamos?** Usando:

ENV 1,5,10,20

SOUND 1,200,-100,5,1

ENV 1,5,10,30

ENV 1,7,3,2

ENV 1,17,30,40

se nos producirán algunos resultados interesantes si somos bastantes rápidos al introducir todas las nuevas envolventes mientras se mantiene el sonido.

Podemos hacer lo mismo con la envolvente de tono. Probemos suerte con:

ENT -1,5,10,20

SOUND 1,200,10000,5,0,1

ENT -1,20,-100,20

ENT -1,30,100,5

Y esto es todo por esta semana. La próxima veremos los ruidos (como si no hubiésemos tenido ya bastantes).

GANA 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANTAL

Porque pretendemos que **AMSTRAD SEMANTAL** sea también vuestra revista, hemos abierto una sección en la que se publicarán los mejores programas originales recibidos en nuestra redacción. Vosotros seréis los encargados de realizar estas páginas, en las que podréis aportar ideas y programas interesantes para otros lectores.

Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANTAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.

- Cada programa debe ir acompañado de un texto explicativo en el cual se incluyan:

- Descripción general del programa.

- Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.

- Instrucciones de manejo.

- Todos estos datos deberán ir escritos a máquina o con letra clara para mayor comprensión del programa.

- No se admitirán programas que contengan caracteres de control, debido a que no son correctamente interpretados por las impresoras.

- En una sola cinta puede introducirse más de un programa.

- Una vez publicado, **AMSTRAD SEMANTAL** abonará al autor del programa de **15.000 a 100.000** pesetas, en concepto de derechos de autor.

- Los autores de los programas seleccionados para su publicación, recibirán una comunicación escrita de ello en un plazo no superior a dos meses a partir de la fecha en que su programa llegue a nuestra redacción.

- **AMSTRAD SEMANTAL** se reserva el derecho de publicación o no del programa.

- Todos los programas recibidos quedarán en poder de **AMSTRAD SEMANTAL**.

- Los programas sospechosos de plagio serán eliminados inmediatamente.

¡ENVIANOS TU PROGRAMA!

Adjuntando los siguientes datos:

Nombre y apellidos,

dirección y teléfono.

Indicando claramente en el sobre:

AMSTRAD SEMANTAL
a HOBBY PRESS, S. A. La Granja, 39
Pol. Ind. Alcobendas (Madrid)

El programa que analizamos en esta ocasión, pretende editar pantalla para después conservarlas en cinta o disco. Mediante su utilización podrá preparar menús informativos, que después pueden ser visualizados en sus propios programas o simplemente utilizarla como diario electrónico, su uso de usted depende. Pero de lo que no cabe duda alguna, es de que aprenderemos algo más sobre matrices y ficheros.

10 Nombre del programa.
20 Se establece el modo de pantalla, se borra ésta y se envía la ejecución a la línea 220.

230 Mediante dos llamadas al firmware logramos habilitar y que sea visible el cursor. Los usuarios del 664 y 6128 pueden sustituirlo si lo desean por Cursor 1,1.

240 Dimensionamos una matriz de 24 por 80. Cada elemento contendrá el carácter correspondiente a la posición dada. Después rellenamos toda la matriz con " ". Por este motivo el programa tarda unos segundos en hacernos caso.

260 Asignamos a la variable FI, fila, el valor 1 y a la variable CO, columna, también 1.

270 Junto con el WEND de la línea 310, obligamos a que se repita ese bloque «while» la tecla pulsada no sea la ENTER. Momento en el que habremos finalizado la edición.

280 Colocamos el cursor en la posición FI,CO y ponemos en TE\$ como tecla pulsada ninguna, esto es, TE\$="".

290 Leemos el teclado mediante la función INKEY\$ y realizamos esta labor tantas veces como sea necesario hasta que se haya detectado la pulsación de una tecla.

300 Interrogamos sobre si el código ASCII del carácter elegido se encuentra en el margen 239-244. En este intervalo se encuentran los caracteres generados por el cursor las teclas del cursor. De ser así llamamos a la subrutina MUEVE CURSOR. Lo único que nos interesa de ésta es que se mueve los cursores arriba, abajo, derecha o izquierda. En el caso contrario enviamos la ejecución a la subrutina ESCRIBE CARACTER, que escribirá el correspondiente carácter en pantalla.

330 Una vez hallamos decidido acabar la edición, se abre el fichero TEXTO, si desea puede cambiar este nombre.

340 Mediante dos bucles anidados y el uso de la sentencia WRITE escribimos los datos en el disco o la cinta.

360 Mandamos la ejecución a la línea 160, en la que se encuentran las instrucciones necesarias para volcar los datos desde el soporte, la cinta o el disco a la pantalla.

Subrutinas:

30' ESCRIBE CARACTER.

40 Si la tecla pulsada es la tecla ENTER devolvemos la ejecución al panorama principal y aquí no ha pasado nada.

50 Cargamos el elemento LE\$(FI,CO) con el carácter elegido.

60 Imprimimos en pantalla este carácter para poder ser visualizado.

70 Incrementamos la variable, CO en una unidad. En el caso de haber alcanzado la columna 81 mandamos la ejecución a la subrutina SALTO DE LINEA. En caso contrario RETURN.

90' SALTO DE LINEA.

100 Ponemos el cursor en la columna 1. Y sólo en el caso de no encontrarnos al final de la página incrementamos FI en 1, es decir, saltamos de línea.

110 RETURN.

120' MUEVE CURSOR.

130 Sumamos a FI, -1 en el caso de haber pulsado cursor arriba y de que nos encontremos en una línea que no sea la 1, téngase en cuenta que la línea 0 no existe. Nos basamos para esto en una de las capacidades lógicas de nuestro CPC. (3=3) es -1 y («HOLA»=«ADIOS») es 0, por ser verdadero y falso respectivamente.

140 Idéntica operación para cursor derecha, izquierda. Después RETURN.

160' LECTURA.

170 Abrimos el fichero «TEXTO» para lectura.

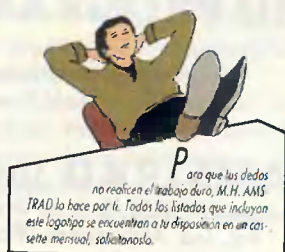
180 Traducido al castellano viene a decir: mientras no estés al final del fichero lee un dato y escríbelo en pantalla.

200 Cerramos el fichero.

210 Devolvemos la ejecución al programa principal.

Una vez que haya editado su pantalla, puede incluir la subrutina LECTURA, sólo ésta, en un programa que usted realice, cuando pretenda visualizar ésta.

¡Ah!, lo olvidaba. La línea 380 es... el fin.



& Juegos ESTRATEGIA

¡Juegos de Simulación Estratégica para poner a prueba su inteligencia y vivir la emoción de situaciones reales!

Un WAR GAME de estrategia que reúne todos los condimentos necesarios para hacer de él un auténtico «plato fuerte», inteligencia, emoción, sorpresa y realismo.

Sólo 1.125pts.
(I.V.A. incluido)

RATAS del DESIERTO



HOBBY PRESS
Para gente inquieta.

um 48, Plus, 128 K

¿Se atreve Ud. a dirigir la Campaña del Desierto y derrotar a Rommel antes de que lo hiciera el General Montgomery en el Alamein?

RATAS del DESIERTO

Operación Norte de Africa

¡Desde uno a tres jugadores!

Disponible para

Spectrum

y

Amstrad

Recorte o copie este cupón y envíelo a Hobby Press, S. A. Apdo. de Correos 54.082 Madrid.
Deseo recibir en mi domicilio, sin gastos de envío alguno por mi parte, la cinta RATAS DEL DESIERTO al precio de 1.125 ptas. (IVA incluido).

Nombre
Dirección
Localidad
Código
El modelo que elijo es para:
☐ Amstrad ☐ Spectrum
La forma de pago elegida es la que señalo con una cruz:
☐ Giro postal n.º ☐ Tarjeta Visa n.º
Pres. S. A. Fecha de caducidad de la tarjeta
Edad
Provincia
Teléfono
Fecha y firma:
Talón nominativo a Hobby

RATAS DEL DESIERTO

La Segunda Guerra Mundial hace tiempo que ha empezado. Nos encontramos en un tiempo y un lugar en el que la Tierra entera es pasto del fantasma de la guerra y del terror.

L

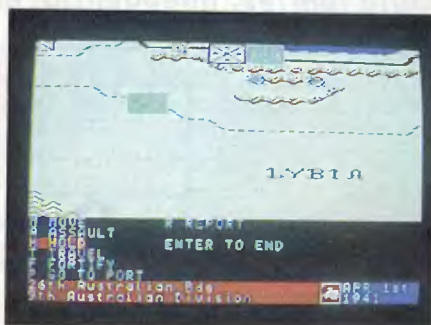
a destrucción y el odio rien a carcajadas entre los hombres, y el teatro bélico se extiende insidiosamente, como un cáncer emponzoñado, por todas partes del mundo.

Hitler estaba ganando en todos los frentes, la sangrienta bandera de la Cruz Gamada ondeaba, orgullosa, en toda Europa.

Alemania trasladó su atención a intentar acabar con el último foco de resistencia: Inglaterra.

Para ello, los nazis necesitaban conseguir dos cosas: petróleo, y liquidar el Imperio Colonial británico, llegar hasta la India y destruirla.

El camino elegido fue a través del Norte de Africa, y el encargado de realizarlo el posteriormente General-feldmarshall (*mariscal*) Erwin Rommel, el militar más experto y astuto que Hitler tenía a su disposición.



Aquí es donde entra en juego el programa de «**Ratas del desierto**», una simulación por ordenador rigurosamente histórica de la Guerra del Desierto, en la cual nosotros nos ponemos al mando de uno de los dos ejércitos, y tratamos de conseguir sus objetivos lo más eficazmente posible. ¿Quién es capaz de rehuir la tentación de cambiar la Historia aunque sea en la pantalla de un **Amstrad**, y ponerse en las botas de Rommel, el Zorro del desierto, o del mariscal Montgomery, nacido para ser su enemigo?

El programa de las «**Ratas del desierto**» es un juego de estrategia militar en su más puro estilo: nada de nubes de marcianitos que tratan de aniquilarnos sin misericordia, nada de crisis nerviosas al mando rebelde de un joystick; aquí reina la serenidad, la posibilidad de sentarse tranquilamente y planear la estrategia adecuada para aniquilar al enemigo sin prisas ni agobios.



Compatible: CPC/464, CPC/664 y CPC/6128

RATAS del DESIERTO

AMSTRAD
CPC/464, CPC/664, CPC/6128



Tenemos todo el Norte de Africa para nosotros solos, encuadrado dentro de un montón de escenarios distintos, que cubren con gran verosimilitud todos los aspectos de la guerra del desierto que tuvo lugar desde la primavera de 1941 hasta la retirada del Afrika Korps a finales de 1942.



En cuanto al aspecto del programa, los lectores podrán juzgar por sí mismos a la vista de las fotos; podemos decir que los autores del programa han preferido representar las unidades que combaten figuradamente, o sea, con cuadrados y rectángulos muy bien dibujados en lugar de tanques o aviones, por ejemplo.

El jugador está asistido en todo momento por una pantalla de mensajes e información, acerca de los diversos aspectos del combate y de las unidades que entran en él.

Los gráficos y el color están muy bien conseguidos, y el movimiento de las unidades a través del Norte de Africa se realiza con suavidad y precisión; el método escogido para presentarlo es mostrar en pantalla una ventana, un trozo del mapa del escenario de la batalla, a través del cual nos desplazamos mediante determinadas teclas de movimiento, que pueden o no arrastrar en su camino a la unidad de combate.

Mister JOYSTICK

Podemos ponernos de parte de los aliados o de las fuerzas del Eje; del bando enemigo siempre se hace cargo del ordenador, con sorprendente eficacia, por otra parte. Juega pero que muy bien; nos costará ganarle.

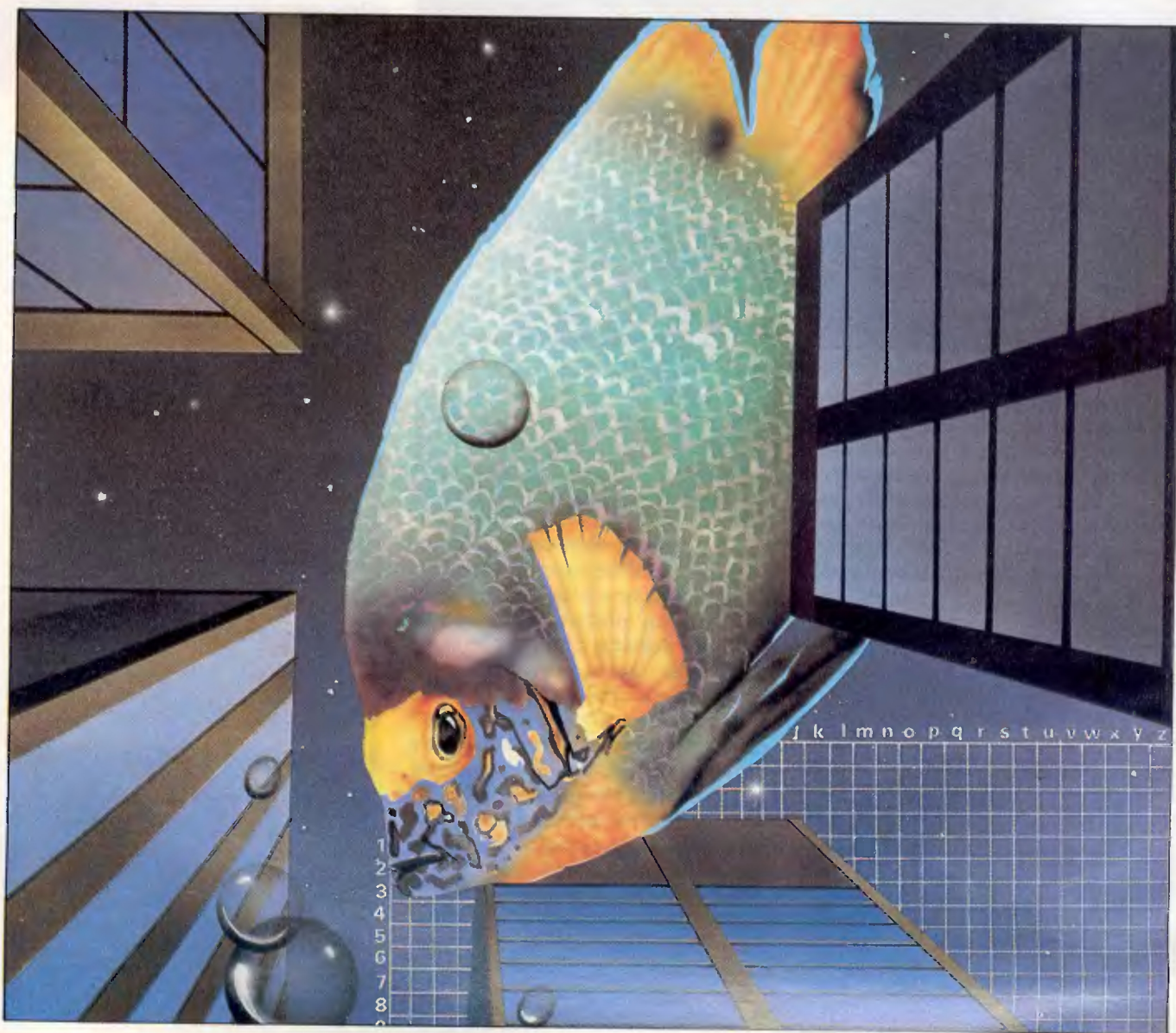
En definitiva, «**Ratas del desierto**» es un juego de la más pura estrategia, no exento de emoción, con un alto nivel de dificultad, unos gráficos y un movimiento bastante logrados y que proporcionará horas y horas de entretenimiento a cualquier persona que se sienta atraída por el desafío intelectual que plantea.



LA RAZON DE SER DE LOS ORDENADORES ES **MASTERCALC**

*A todos aquellos que se preguntan si un ordenador puede servir para algo: ¡va por ustedes!
A mí, como usuario de ordenadores además de apasionado por la informática, no me cabe duda alguna de que habría que levantar una estatua a los creadores de un programa llamado Visicalc, firmado por la «multitud agradecida», o algo igualmente entrañable.*

La razón es muy simple: los que hicieron Visicalc convencieron a todo el mundo de que los ordenadores servían para algo, ¡vaya que si servían!, y son los indirectos responsables de la existencia del programa Mastercalc, nuestra estrella de hoy en Banco de Pruebas. Mastercalc, como Visicalc, es una hoja de cálculo, y se ha seguido la tradición de que el nombre del programa acabe en «calc»; hasta ese punto llega la influencia del «original». Bien, tal vez sea ya el momento de explicar un poco, para el que le sea útil, lo que es una hoja de cálculo.





Se trata de una hoja de papel cuadriculada, pero computarizada. En términos informáticos, una hoja de cálculo es una matriz bidimensional, esto es, una «**cosa**» compuesta por filas y columnas, inicialmente vacías, de unas dimensiones dadas.

A los huecos que forman las filas y columnas se les llaman «**celdillas**», y se pueden meter cosas en ellas.

¿Que qué cosas?, pues números, texto y fórmulas que relacionan ambas.

La utilidad de las hojas de cálculo estriba en que uno puede responderse rápidamente a la pregunta de «**qué pasaría si...**».

Es decir, si estoy calculando los costes de un proyecto o de una empresa, puedo averiguar en cualquier momento cómo cada uno de los factores involucrados afectan al conjunto. Por ejemplo, **¿cómo afecta al coste total si en vez de contratar a dos empleados contrato sólo a uno e invierto un poco más en maquinaria?**

Espero que, al leer esto, más de un atribulado empresario «**calculador**» comience a frotarse las manos de anticipado placer, pero esto no es todo: los cambios se reflejan instantáneamente en los resultados finales del cálculo que estemos haciendo, porque cada vez que introducimos un dato, el programa recalcula todas las celdas donde se encuentre información relacionada con lo que acabamos de variar.

Excuso decir que la mayoría de estos programas, como es el caso de Mastercalc, permite visualizar la información y su evolución gráficamente, en general en forma de diagramas de barras o de cualquier otro tipo.

Naturalmente, todo esto puede ser trasladado a papel mediante una impresora.

En resumen, las hojas de cálculo son tal vez la aplicación que demuestra de forma más obvia la gran ventaja en tiempo y dinero que puede suponer el uso inteligente de un ordenador, de su **Amstrad**.

EJEMPLO MICROHOBBY AMSTRAD					
	1	2	3	4	5
	Ene....	Abr....	Jul....	Oct....	TOTALES
Fuel-oil	345,000	325,000	326,100	331,000	3,960,600
Gas-oil A	152,000	151,500	126,000	134,000	1,660,500
Gas-oil B	60,000	60,000	60,000	60,000	720,000
Grasa ind.	7,245	7,245	7,245	7,245	86,940
Electrici.	320,000	320,000	320,000	320,000	3,840,000
Mano obra	547,000	547,000	547,000	547,000	6,564,000
Alquiler	8,255	8,255	8,255	8,255	99,060
Telefono	7,330	7,330	7,330	7,330	87,960
Mat.Oficina	12,916	12,916	12,916	12,916	154,992
COSTES GR.	1,459,746	1,439,246	1,414,846	1,427,746	17,174,052
Mat. Primas	840,000	840,000	840,000	840,000	10,080,000
Envases	600,000	600,000	600,000	600,000	7,200,000
Embalajes	170,000	170,000	170,000	170,000	2,040,000
Aditivo A	28,933	28,933	28,933	28,933	347,196
Aditivo B	38,400	38,400	38,400	38,400	460,800
Aditivo C	19,200	19,200	19,200	19,200	230,400

CTRL K for menu

Uno de los programas de gestión más utilizados por los poseedores de ordenadores es la Hoja de Cálculo. Como su nombre indica, una Hoja de Cálculo es una herramienta que permite a través de unos datos dados, obtener unos resultados definidos mediante unas fórmulas dadas previamente. En tareas estadísticas, contables, presupuestarias, etc..., puede resultar de una gran utilidad.

1. Inicio del programa

Una vez cargado el programa, nos sale el menú principal. En él están las opciones principales del programa, que figuran en el cuadro 1. También se puede acceder a este menú pulsando simultáneamente las teclas CTRL y X. En éste están las opciones principales de carga, grabación, catálogo del disco, cambio a cinta, ir al BASIC y la de elegir el color de la pantalla y de la tinta, que resulta muy cómoda dado que no usa números sino que basta con pulsar P para cambiar el color del papel, I para cambiar el de la tinta, y B para cambiar el del borde.

Pulsando CTRL y K, nos sale la pantalla que corresponde al cuadro 2. Esta, simplemente nos recuerda la totalidad de opciones que se pueden usar a lo largo del programa.

2. Creación de una hoja

Al crear una nueva hoja, lo primero que pide es el número de filas y de columnas que se desean, y luego pasa directamente a la pantalla principal, donde éstas se ven numeradas. Si se han definido más de 9 filas o 20 columnas, éstas no salen en pantalla al principio, pero se pueden consultar llevando el cursor hasta la última fila o columna y pulsando SHIFT y las teclas de movimiento del cursor. Las celdas en principio tienen una longitud de 8 ca-

racteres y ésta se puede modificar mediante CTRL y A. Esta opción también nos permite modificar el número de filas o columnas y el nombre del plan en cualquier momento.

3. Movimiento del cursor

Ya con la hoja en blanco en la pantalla, el cursor se puede mover por todas las celdas de la pantalla mediante las teclas de cursor. Pero, si tenemos más filas o columnas de las que salen en la pantalla, hay diversas maneras de acceder a ellas. La primera y más sencilla, es situar el cursor en la última fila o columna y pulsar SHIFT y la tecla de movimiento correspondiente. Esto hace que en pantalla salgan las siguientes 7 columnas o 20 filas. Pero puede que se desee ver las cinco primeras columnas y las dos últimas. En este caso, con la opción 0 se puede intercalar en cualquier lugar una columna determinada. Resulta muy útil, para tener a la vista la columna de los totales, por ejemplo. Otras opciones realizan movimientos determinados. Si estamos dando datos por filas, podemos hacer que el cursor avance automáticamente hacia la derecha con CTRL R, y si lo estamos haciendo hacia abajo, igual con CTRL D. Para llevar rápidamente el cursor a la fila 1 columna 1, CTRL H. El resto de las opciones de movimiento de cursor están explicadas en el cuadro 2.

4. Ventanas

El programa asume que la primera fila y la primera columna están destinadas a contener los nombres de las celdas. Por esto, cuando se cambia de pantalla, cambian sólo a partir de las segundas. Aparte de esto, podemos tener en pantalla las columnas que nos interesen, aunque estén saltadas (Opción 0). Desgraciadamente esto no se puede hacer con las filas. Si deseamos una fila determinada, ten-

dremos que tener también las anteriores o posteriores. Esta incomodidad se puede soslayar usando la otra ventana de que dispone el programa. Se puede dividir la pantalla en dos ventanas, aunque el cursor sólo pueda operar en una de ellas, con la opción S. Entonces aparece una pantalla con las mismas columnas aunque con la mitad de filas y otra idéntica debajo. El cursor operará en la superior, mientras que no se utilice la opción W, que cambiará a la inferior.

5. Introducción de datos

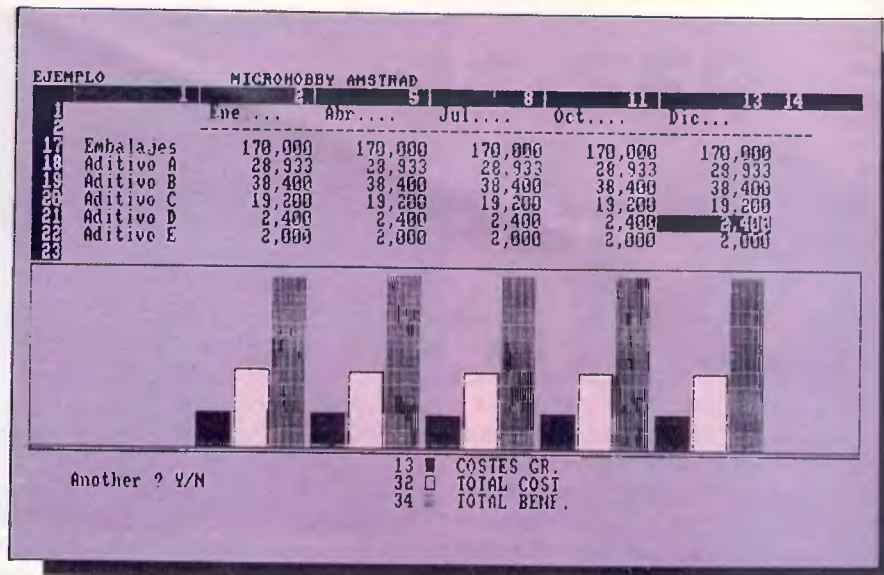
Para introducir nombres en las celdas, una vez colocado el cursor en la celda deseada, hay que pulsar CTRL y ". Para introducir números, basta con teclearlos con o sin signo. Si bien la longitud inicial de las celdas es de ocho caracteres, el número puede tener más de ocho cifras, aunque no saldrá completo en la pantalla (*ni en la impresora*), pero al hacer los cálculos tomará en cuenta el número entero. De todas maneras, como se puede modificar la longitud de las celdas, es mejor prever las cifras que se van a utilizar y modificar la longitud para que sobren uno o dos espacios. La mayor cantidad de cifras que puede manejar sin peligro de error es de dieciséis cifras y el menor número 0,0000001 ya que el mayor número de decimales que soporta es 7. Atención a una cosa: Si las operaciones van a dar lugar a decimales, hay que definir el número de decimales que queremos, mediante CTRL y A, ya que si no, el programa redondeará los cálculos al número más próximo. Para borrar alguna celda, si se va a introducir el mismo tipo de datos, números, textos o fórmulas, basta con volverlos a introducir. Pero si no es así, hay que anular el dato con la opción CTRL y Z. Esta opción también sirve para borrar toda una columna o fila.

Cuando tengamos que repetir un valor varias veces, se puede utilizar SHIFT &, que copiará el valor en todas las filas o columnas por las que pase el cursor.

6. Fórmulas

En cualquier celda podemos poner una fórmula. Esta es una de las opciones más interesantes de la hoja. Podemos hacer referencia en las fórmulas a valores de otras celdas. Así podemos tener, por ejemplo, una fila o columna con el porcentaje de el valor de otra, o con totales, etc...

Previamente hay que definirla con la opción CTRL y F. Se le da un número de referencia, entre 1 y 99, y se puede hacer referencia a filas y columnas simplemente poniendo R3 (*fila 3*), o C4 (*columna 4*) en las fórmulas. Las operaciones tienen la misma prioridad que en BASIC a la hora de poner paréntesis.



Otras opciones importantes son T y CTRL T. T permite definir celdas que serán el subtotal o total de los valores de las celdas anteriores, en filas o columnas. Y CTRL T muestra las celdas que contienen fórmulas o totales, con el número de fórmula o el tipo de total que contienen.

Cada vez que se modifique algún dato que influya en fórmulas o totales, hay que utilizar la opción CTRL y C, para que la hoja recalculé los valores de dichas fórmulas o totales. Hay que tener en cuenta, que si hay alguna fórmula o total que haga referencia a celdas que estén situadas más abajo o a la derecha, habrá que hacer DOS veces este proceso para que los cálculos sean correctos.

7. Gráficos

La opción CTRL G realiza un gráfico de barras de un máximo de tres filas y las columnas que estén en pantalla en ese momento. No se pueden hacer gráficos de columnas. Los gráficos resultan un poco pobres y la única manera de imprimirlos es tener la suerte de poseer una impresora EPSON o compatible, (*NO LA DMP-1*). Si es así, CTRL O hace una copia en alta resolución de lo que haya en la pantalla (*Tarda unos 3 minutos*). Esta opción no se muestra en menú, al no valer para todas las impresoras. Naturalmente esta opción vale también para imprimir cualquier otra cosa que esté en la pantalla, pero realmente es más rápido y cómodo utilizar la opción de impresión, que es la adecuada.

8. Impresión

Se puede listar el plan por cualquier impresora mediante CTRL P. Podemos elegir los códigos de tipo de letra, anchura, etc..., y a partir de qué fila empieza a imprimir. Esto es útil, dado que si el plan tiene demasiadas, es probable que no nos quepa en la hoja, y haya que imprimirlo en dos o más partes. Podemos

elegir primera y última fila, pero las columnas dependen de la anchura en caracteres que le demos al programa cuando nos lo pida. El programa imprime los números de fila y columna si queremos, y luego a partir de la primera columna que salga en pantalla, todas las que quepan.

9. Instrucciones y manejo

El manual de instrucciones resulta claro y preciso..., siempre que sepamos inglés. No obstante, resulta sumamente sencillo. La rapidez no es como para marearse desde luego, pero resulta más que suficiente a no ser que el plan sea realmente grande. La mayor lentitud estriba en que cada vez que se recalcula o se cambian filas o columnas rehace el plan entero. En cuanto a la memoria, en todo momento podemos saber la que no queda con la opción CTRL S. En principio, el programa deja libres 26823 bytes. Curiosamente, se



EJEMPLO

MICROHOBBY AMSTRAD

	Ago....	Sep....	Oct....	Nov....	Dic....
Fuel-oil	325,000	341,000	331,000	322,000	323,000
Gas-oil A	130,000	157,000	122,000	143,000	123,000
Gas-oil B	60,000	60,000	60,000	60,000	60,000
Grasa ind.	7,245	7,245	7,245	7,245	7,245
Electrici.	320,000	322,000	345,000	337,000	349,000
Mano obra	547,000	547,000	547,000	547,000	547,000
Alquiler	8,255	8,255	8,255	8,255	8,255

Another " Y/N

 3 ■ Fuel-oil
 4 □ Gas-oil A
 7 ▼ Electrici.

EJEMPLO

MICROHOBBY AMSTRAD

	Ene....	Feb....	Mar....	Abr....	May....
Fuel-oil	345,000	340,000	350,000	325,000	300,000
Gas-oil A	157,000	147,000	143,000	151,500	135,000
Gas-oil B	60,000	60,000	60,000	60,000	60,000
Grasa ind.	7,245	7,245	7,245	7,245	7,245
Electrici.	320,000	320,000	320,000	320,000	320,000
Mano obra	547,000	547,000	547,000	547,000	547,000
Alquiler	8,255	8,255	8,255	8,255	8,255
Telefona	7,330	7,330	7,330	7,330	7,330
Mat.Oficina	12,916	12,916	12,916	12,916	12,916
COSTES GR.	1,459,746	1,445,746	1,471,746	1,439,246	1,397,746
Mat. Primas	840,000	840,000	840,000	840,000	840,000
Envases	600,000	600,000	600,000	600,000	600,000
Embalajes	170,000	170,000	170,000	170,000	170,000
Aditivo A	20,933	20,933	20,933	20,933	20,933
Aditivo B	38,400	38,400	38,400	38,400	38,400
Aditivo C	19,200	19,200	19,200	19,200	19,200

+ CTRL K for menu

ocupa más memoria si la hoja tiene 10 filas y 20 columnas, por ejemplo, que si la hoja tiene 20 filas y 10 columnas.

10. Conclusiones

Antes de ponerse a trabajar con él, es conveniente adquirir un poco de práctica. Se deben tener en cuenta las limitaciones del mismo, por ejemplo si vamos a necesitar gráficos y de qué los vamos a necesitar, cuántas filas y columnas serán necesarias, etc... Es francamente útil a la hora de realizar cálculos o comparaciones económicas, llevar control de gastos y en cualquier actividad donde se manejen números y cálculos. Las posibilidades del mismo van en función de cómo se utilice, porque realmente es una hoja de cálculo con potencia suficiente para dar respuesta a las necesidades de casi todo poseedor de un **Ams-trad**.

CUADROS DE COMANDOS

MENU PRINCIPAL

Se selecciona pulsando simultáneamente [CTRL] y [X]:

N: Crear un nuevo plan.

L: Cargar un plan del disco/cinta.

S: Grabar un plan del disco/cinta.

C: Proporciona el directorio del disco.

R: Retorna al plan anterior.

T: Selecciona la cinta para leer y grabar.

D: Selecciona el disco para leer y grabar.

*: Elegir los colores de fondo, borde y tinta.

X: Volver al Basic.

MENU DE COMANDOS

Se seleccionan pulsando la tecla correspondiente:

C: Pide n.º de columna y la muestra en pantalla.

F: Sitúa el cursor en la fila de arriba, columna izquierda.

H: Sitúa el cursor en la fila de arriba, columna izquierda.

L: Sitúa el cursor en la columna izquierda.

M: Intercambia el Modo de pantalla entre 1 y 2.

O: Pide n.º de columna y muestra ésta y las siguientes.

R: Pide n.º de fila y la muestra en la pantalla.

S: Activa y desactiva la segunda ventana.

T: Permite definir los totales y subtotales.

W: Selecciona la ventana superior o inferior (ver S).

&: Repite el valor introducido anteriormente.

=: Mueve el cursor a la primera fila.

": Permite la entrada de texto.

+, - o Números: Permite la entrada de valores.

Teclas de cursor... Mueven el cursor por el plan.

Pulsando [CTRL] y la tecla correspondiente:

A: Altera el tamaño, título y formato del plan.

C: Realiza los cálculos de las fórmulas.

D: Mueve automáticamente el cursor hacia abajo.

F: Crea o borra fórmulas.

G: Realiza un gráfico de barras de 3 filas.

H: Sitúa el cursor en la fila 1, columna 1.

K: Muestra estas opciones.

P: Imprime el plan.

R: Mueve automáticamente el cursor hacia la derecha.

S: Muestra la capacidad libre de memoria.

T: Muestra en el plan las casillas que usan fórmulas.

X: Muestra el menú principal.

Z: Borra el valor de una celda, fila o columna.

O: Sólo funciona en determinadas impresoras. Saca una copia impresa de la pantalla. Muy útil para imprimir los gráficos. No funciona en la DPM1.

[SHIFT] + teclas de movimiento de cursor: Mueven la ventana de texto actual.

TURBOGRAF

Este es un programa destinado al cálculo integral —mediante el método Simpson— y representación gráfica de funciones, con un original sistema de introducción de las mismas, diseñado de una manera clara y agradable. Será una inestimable ayuda para estudiantes y profesionales de la rama de ciencias, sin olvidarse de cualquier persona deseosa de adentrarse en los secretos de la programación.

Programa realizado por: **José Vicente Martí**

Se trata de un programa que permite representar gráficamente cualquier función previamente definida y hallar los siguientes datos:

- Imagen en un punto.
- Derivada numérica en un punto.
- Integral definida entre dos puntos.
- Ceros de la función.

La presentación se compone de cuatro ventanas divididas en tres grupos:

— Entrada de datos y salida de resultados: Ventana que ocupa la parte superior izquierda de la pantalla.

— Menú y datos de la gráfica: Parte derecha de la pantalla.

— Gráfica del intervalo escogido: En la parte inferior izquierda de la pantalla.

El modo de pantalla utilizado es el de 80 columnas de forma que la cantidad de información que proporciona la pantalla es máxima y habiendo escogido los colores para que resulte perfectamente legible incluso en un monitor policromo.

Funcionamiento del programa:

Tras haber cargado el programa en memoria y tecleado RUN aparecerá la distribución de pantalla con el menú como única información. Pulso el mismo, en vídeo inverso, aparece el mensaje **«ELIGE OPCIÓN»**. Pulsando un número del 1 al 7 este mensaje será cambiado por **«OP-**

CIÓN n SEGURO (S/N) ? al que deberemos responder con «S» si deseamos pasar a la opción señalada y con «N» si queremos volver a elegir opción. Esta medida tiene su razón de ser en que no podamos tomar un camino equivocado y tener que esperar o introducir datos para tener que acceder nuevamente al menú.

Una vez en la opción escogida, tendremos que atenernos a las características de cada una de ellas.

1. REPRESENTAR INTERVALO: En la ventana de entrada de datos aparece el título de la opción escogida (*esto sucede en todas las opciones por lo que a partir de ahora se sobreentiende*). También podemos ver la inscripción: **«a?»**.

En esta opción se pretende representar gráficamente los valores que toma la función en un determinado intervalo (a,b) , es decir, que vamos a dar valores a la variable (*que llamaremos «x»*) desde a hasta b y hallar los valores de la función representándolos. Así pues, los datos que se nos piden son a y b .

Una vez introducidos, aparece una nueva pregunta: **«NUMERO DE PUNTOS ?»** que significa cuántos puntos vamos a calcular o en cuántas partes vamos a dividir el intervalo (a,b) para representarlo. Este número depende de lo complicada o **«agreste»** que pueda ser la gráfica. Normalmente bastarán 300 puntos, que es el valor que tomará esta variable si pulsamos simplemente **«ENTER»**.

REFORMA LA LINEA 88 COMO SIGUE:
88 DEF FNy(x)=IFUNCION1
LUEGO PULSA [ENTER] PEQUEÑA
88 DEF:DEF FNy(x)=EXP(x)

REPRESENTACION/ENTRADA DE DATOS
MENÚ Y 1
MÁXIMO Y 2
NÚMERO DE PUNTOS Y 10

1 - REPRESENTAR UN INTERVALO
2 - HALLAR IMAGEN DE UN PUNTO
3 - HALLAR DERIVADA EN UN PUNTO
4 - HALLAR INTEGRAL DEFINIDA
5 - HALLAR CEROS DE LA FUNCIÓN
6 - INTRODUCIR FUNCIÓN
7 - FIN

REPRESENTACION/ENTRADA DE DATOS
MENÚ Y 10
MÁXIMO Y 15
NÚMERO DE PUNTOS Y 10

OPCIÓN 10
ENTRADA DE DATOS: a=0.000000
b=1.000000
NÚMERO DE PUNTOS: 10
MÁXIMO: 15

1 - REPRESENTAR UN INTERVALO
2 - HALLAR IMAGEN DE UN PUNTO
3 - HALLAR DERIVADA EN UN PUNTO
4 - HALLAR INTEGRAL DEFINIDA
5 - HALLAR CEROS DE LA FUNCIÓN
6 - INTRODUCIR FUNCIÓN
7 - FIN

SUBROUTINAS PRINCIPALES

80	Definición de la función
100-250	Bucle principal. Menú
260-280	Tratamiento de respuestas S/N
390-500	Búsqueda del máximo y el mínimo en el intervalo a representar
510-630	Conversión de coordenadas y cálculo de origen de coordenadas
640-770	Trazado de la gráfica
760	Realiza el rellenado de la superficie de integración en caso de que ésta se realice
800-870	Cálculo de la imagen de un punto
880-990	Cálculo de la derivada en un punto
1030-1160	Cálculo de la integral en un intervalo
1170-1240	Representar intervalo de integración gráficamente
1250-1440	Hallar los ceros de la función
1450-1550	Introducción y modificación de la función
1630-1880	Inicialización

VARIABLES PRINCIPALES

— tes:	Respuesta a una pregunta S/N
— xmi, xma	Mínimo y máximo del intervalo a representar. Valores proporcionados por el usuario
— n	Número de puntos a representar
—rs, rt	Mínimo y máximo relativos (utilizados para la búsqueda de los absolutos)
— insma, insmi	Puntos de inspección en la búsqueda del máximo y el mínimo de $f(x)$ en el intervalo (xmi, xma)
— st	Incremento de búsqueda
orx, ory	Coordenadas del origen en la pantalla
— ox, oy	Coordenadas reales del origen
— cx, cy	Factores de conversión de coordenadas
— px, py	Coordenadas del punto a representar en la pantalla
— integ	Indica si el aparato de representación se utiliza como subrutina del de integración
— x	Punto del que se halla la imagen
— vert	Valor del incremento de y al hallar la derivada
— deriv	Valor de la derivada en el punto x
— a, b	Extremos de integración
— suma	Variable donde se va acumulando el valor de la integral
— ceros	Indicador de que el apartado de hallar la derivada se utiliza como subrutina de cálculo de los ceros de la función
— count	Limitador de interacciones para CEROS. Indica cuándo la semilla utilizada es inútil
— con	Contador del número de puntos representados para hacer el sombreado de la zona de integración

Una vez cumplimentados los datos se iniciará la representación apareciendo en primer lugar los datos de la gráfica que son:

- x MINIMO: Valor mínimo de las x, es decir, a
- x MAXIMO: Valor máximo de las x, es decir, b
- y MINIMO: Valor mínimo que toma la función en el intervalo
- y MAXIMO: Valor máximo que toma la función en el intervalo
- x ORIGEN: Coordenada x del punto que aparece como origen en el dibujo
- y ORIGEN: Coordenada y del punto que aparece como origen en el dibujo.

Acto seguido, empezará a dibujar la gráfica.

EJEMPLO: Supongamos que representamos la función $\cos(x)$ en el intervalo (0-360) en grados sexagesimales, es decir, en modo DEG:

xMINIMO: 0 xMAXIMO: 360
yMINIMO: -1 yMAXIMO: 1
xORIGEN: 0 yORIGEN: 0

2. HALLAR IMAGEN EN UN PUNTO: Se nos pide el dato «PUNTO x?» al que hay que responder con el valor que queremos darle al argumento de la función. Al momen-



to aparecerá el valor « $f(x)=$ » (resultado).

EJEMPLO: Siguiendo con la función $\cos(x)$ que mantendremos en todos los ejemplos:

PUNTO $x=180$ $f(x)=-1$

3. Hallar DERIVADA EN UN PUNTO: Se utiliza el método de definición geométrica de derivada, a saber:

Para ello debemos proporcionar el valor del punto x en el que queremos hallar la derivada y el valor de incremento de x que adopta el valor por defecto (si pulsamos ENTER sin entrar ningún valor) de 0.0001

EJEMPLO:

PUNTO $x=90$ $f'(x)=-1$

Serie ORO

4. INTEGRAL DEFINIDA: Mediante el método de los trapecios que consiste en dividir en un número determinado de trapecios la superficie de integración definida por el eje de las x y la gráfica de la función entre los extremos de integración. Posteriormente se suman las superficies de los rectángulos que constituirán el valor de la integral definida.

Por lo tanto, necesitamos los datos: a, b y número de trapecios. Este último toma por defecto el valor de 100.

Una vez calculada la integral, aparecerá el mensaje «REPRESENTAR (S/N)?» En el caso de que respondamos afirmativamente, representará el intervalo de integración sombreado la superficie a que equivale la integral.

NOTA: Si usamos las funciones trigonométricas en el modo DEG, los resultados de la integración no coincidirán con los valores reales.

5. HALLAR CEROS DE LA FUNCIÓN: Utilizando el método de derivaciones sucesivas o de las tangentes.

Necesitamos una semilla o número a partir del cual iniciaremos la búsqueda de los ceros. También hay que proporcionarle un intervalo en el cual podamos considerar que el valor de la función es cero, es decir, si $f(x) \leq r$, siendo r el valor requerido, se considera que $f(x)=0$. Esta variable toma el valor por defecto de 0.000001.

6. INTRODUCIR LA FUNCIÓN: Al elegir esta opción la pantalla pasa a MODE 1 y aparecen instrucciones para introducir la función, así como la línea 80 que es donde se define la función. Después de alterada convenientemente basta pulsar (RUN ENTER)* para que el programa vuelva a funcionar.

7. FIN: Al elegir esta opción abandonamos el programa. Igual que en la opción anterior, basta pulsar (RUN ENTER)* para volver al mismo.

* En lugar de RUN ENTER, pulsar ENTER pequeña.


```

80 DEG:DEF FNY(X)=EXP(X)
90 GOSUB 1630
100 LOCATE #2,5,1:PRINT #2,"1 - REP
RESENTAR UN INTERVALO"
110 LOCATE #2,5,3:PRINT #2,"2 - HAL
LAR IMAGEN DE UN PUNTO"
120 LOCATE #2,5,5:PRINT #2,"3 - HAL
LAR DERIVADA EN UN PUNTO"
130 LOCATE #2,5,7:PRINT #2,"4 - HAL
LAR INTEGRAL DEFINIDA"
140 LOCATE #2,5,9:PRINT #2,"5 - HAL
LAR CEROS DE LA FUNCION"
150 LOCATE #2,5,11:PRINT #2,"6 - IN
TRODUCIR FUNCION"
160 LOCATE #2,5,13:PRINT #2,"7 - FI
N"
170 LOCATE #2,19,13:PAPER #2,1:FEN
#2,0:PRINT #2,"ELIGE OPCION:"
180 AS=INKEY$:IF VAL(AS)<1 OR VAL(A
S)>7 THEN 180
190 A=VAL(AS)
200 LOCATE #2,16,13:PRINT #2,"OPCIO
N:"A: (S/N)?:PAPER #2,0:FEN #2,1
210 AS=INKEY$:IF UPPER$(AS)<>"S" AN
D UPPER$(AS)<>"N" THEN 210
220 LOCATE #2,16,13:PRINT #2,SPACE$
(15)
230 IF UPPER$(AS)="N" THEN 170
240 ON A GOTO 320,830,910,1030,1280
,1480,1590
250 END
260 AS=INKEY$:IF UPPER$(AS)<>"S" AN
D UPPER$(AS)<>"N" THEN 260
270 IF UPPER$(AS)="S" THEN tes=1 EL
SE tes=0
280 RETURN
290
-----
300 REM --- REPRESENTA UN INTERVALO
310
-----
320 CLS #1:INTG=0:PRINT #1
330 PRINT #1," REPRESENTACION:EN
TRADA DE DATOS"
340 PRINT #1:INPUT #1," MINIMO ":x
mi
350 PRINT #1:INPUT #1," MAXIMO ":x
ma
360 PRINT #1:INPUT #1," NUMERO DE
PUNTOS ":n
370 IF n=0 THEN n=300:LOCATE #1,22,
8:PRINT #1,"300"
380 IF xmi>xma THEN 320
390 insmi=xmi:insma=xma: ENTRADA I
NTEGRAL
400 rs=xmi:rt=xmi
410 FOR i=1 TO 4
420 IF i=3 THEN insmi=xmi:insma=xma
430 st=ABS(insma-insmi)/30:b=FNY(in
smi):d=FNY(insmi)
440 FOR j=insmi TO insma STEP st
450 IF i>=3 THEN c=MAX(FNY(j),d):IF
c>d THEN rt=j:d=c
460 IF i<3 THEN a=MIN(FNY(j),b):IF
a<b THEN rs=j:b=a
470 NEXT
480 insmi=rs-st/2:insma=rs+st/2:IF
i>=3 THEN insmi=rt-st/2:insma=rt+st
/2
490 NEXT
500 ymi=a:yma=c
510 CLS #3
520 PRINT #3," DATOS GR
AFICA"
530 LOCATE #3,2,3:PRINT #3,"XMINIMO
":ROUND(xmi,3):LOCATE #3,20,3:PRIN
T #3,"XMAXIMO":ROUND(xma,3)
540 LOCATE #3,2,5:PRINT #3,"YMINIMO
":ROUND(ymi,3):LOCATE #3,20,5:PRIN
T #3,"YMAXIMO":ROUND(yma,3)
550 ORX=XMI*290/ABS(XMA-XMI)+23
560 IF XMI>0 OR XMA<0 THEN ORX=23
570 ORY=-YMI*180/ABS(YMA-YMI)+38
580 IF YMI>0 OR YMA<0 THEN ORY=38
590 ORX=0:ORY=0
600 IF XMI>0 OR XMA<0 THEN ORX=XMI
610 IF YMI>0 OR YMA<0 THEN ORY=YMI
620 LOCATE #3,2,7:PRINT #3,"X:ORIGEN
":ROUND(ORX,3)
630 LOCATE #3,20,7:PRINT #3,"Y:ORIGE
N":ROUND(ORY,3)
640 CLG(0):ORIGIN ORX,ORY,16,312,32
,224
650 CX=290/ABS(XMA-XMI):CY=180/ABS(
YMA-YMI)
660 MOVE 0,0:DRAW -320,0,1:DRAW 6
40,0: EJE X
670 MOVE 0,0:DRAW 0,-200,1:DRAW 0
,400: EJE Y
680 CON=0
690 FOR X=XMI TO XMA STEP ABS(XMA-X
MI)/n

```

```

700 CON=CON+1
710 PX=X*CX
720 IF XMI>0 OR XMA<0 THEN PX=(X-X
MI)*CX
730 PY=FNY(X)*CY
740 IF YMI>0 OR YMA<0 THEN PY=(FNY(
X)-YMI)*CY
750 PLOT PX,PY
760 IF INTG=1 THEN IF CON MOD (n/5
0)=0 AND X>A1 AND X<B1 THEN MOVE PX
,0:DRAW PX,PY
770 NEXT
780 IF INTG=1 THEN RETURN
790 GOTO 170
800 REM -----
810 REM = CALCULA IMAGEN EN UN PUN
TO =
820 REM -----
830 CLS #1:PRINT #1:PRINT #1:PRINT #1
," CALCULA IMAGEN EN UN PUNTO"
840 PRINT #1:INPUT #1," PUNTO X ":x
850 Y=FNY(X)
860 PRINT #1:PRINT #1," f("x;")
":y
870 GOTO 170
880
-----
890 REM = CALCULA DERIVADA EN UN PUN
TO =
900 REM -----
910 CLS #1:PRINT #1:PRINT #1," C
ALCULA DERIVADA EN EL PUNTO X":CERO
S=0
920 PRINT #1:INPUT #1," PUNTO X ":x
930 PRINT #1:INPUT #1," INCREMENTO
(i=0.0001)":i
940 IF i=0 THEN i=0.0001:LOCATE #1,
28,6:PRINT #1,"0.0001"
950 VERT=FNY(X+i)-FNY(X)
960 DERIV=VERT/I
970 IF CEROS=1 THEN RETURN
980 PRINT #1:PRINT #1," f'("x;")
":ROUND(DERIV,5)
990 GOTO 170
1000
-----
1010 REM = CALCULA INTEGRAL DEFINID
A =
1020 REM -----
1030 CLS #1:PRINT #1:PRINT #1," I
NTEGRAL DEFINIDA ENTRE a Y b"
1040 PRINT #1:INPUT #1," a ":a
1050 LOCATE #1,20,4:INPUT #1,"b ":b
1060 IF a>b THEN 1030
1070 PRINT #1:INPUT #1," NUMERO DE
INTERVALOS":n
1080 IF n=0 THEN n=100:LOCATE #1,23
,6:PRINT #1,n
1090 SUMA=0:st=ABS(b-a)/n
1100 CON=0
1110 FOR i=a TO b-(st/2) STEP st
1120 CON=CON+1
1130 SUMA=SUMA+(FNY(i)+FNY(i+st))/2
*st
1140 LOCATE #1,30,6:PRINT #1,CON
1150 NEXT
1160 PRINT #1:PRINT #1," INTEGRAL="
:ROUND(SUMA,3)
1170 LOCATE #1,22,8:PRINT #1,"REPRE
SENTAR(S/N)?"
1180 GOSUB 260:LOCATE #1,22,8:PRINT
#1,SPACE$(17)
1190 IF tes=0 THEN 170
1200 XMI=A-ABS(B-A)/10
1210 XMA=B+ABS(B-A)/10
1220 INTG=1:A1=A:B1=B:n=300
1230 CLG(0):CLS #3:GOSUB 390
1240 GOTO 170
1250
-----
1260 REM = HALLAR CEROS DE LA FUNCIO
N =
1270 REM -----
1280 CLS #1:PRINT #1:PRINT #1,"
CEROS DE LA FUNCION":COUNT=0
1290 PRINT #1:INPUT #1," SEMILLA ":
S
1300 PRINT #1:INPUT #1," PRECISION
(p=0.000001)":p
1310 IF p=0 THEN p=0.000001:LOCATE
#1,30,6:PRINT #1,"0.000001"
1320 PRINT #1:PRINT #1," SACAR VAL
ORES INTERMEDIOS (S/N)?"
1330 GOSUB 260:LOCATE #1,3,8:PRINT
#1,SPACE$(33)
1340 PRES=0:IF tes=1 THEN PRES=1
1350 WHILE ABS(FNY(S))>p

```

Serie ORO

```

1360 COUNT=COUNT+1
1370 CEROS=1:X=S:I=0.0001:GOSUB 950
1380 IF DERIV=0 THEN COUNT=101:GOTO
1410
1390 S=S-FNY(S)/DERIV
1400 IF PRES=1 THEN LOCATE #1,10,8:
PRINT #1,S
1410 IF COUNT>100 THEN LOCATE #1,10
,8:PRINT #1,"SEMILLA INUTIL":GOTO 1
70
1420 WEND
1430 LOCATE #1,2,8:PRINT #1,"CERO="
:ROUND(S,5):SPACE$(15)
1440 GOTO 170
1450
-----
1460 REM = INTRODUCCION DE LA FUNCIO
N =
1470 REM -----
1480 CLS #2
1490 MODE 1:PAPER 0:FEN 1
1500 PRINT:PRINT:PRINT "REFORMA LA
LINEA 80 COMO SIGUE:"
1510 PRINT:PRINT:PRINT " 80
DEF FNY(X)=(FUNCION)"
1520 PRINT:PRINT:PRINT "LUEGO PULSA
ENTER1 PEQUE:CHR$(200):"A"
1530 PRINT:PRINT
1540 EDIT 80
1550 END
1560
-----
1570 REM = FINAL 170R
1580
-----
1590 MODE 1:PAPER 0:FEN 1:END
1600 REM -----
1610 REM = PREPARAR PANTALLA
1620 REM -----
1630 MODE 2:INK 0,1:BORDER 1:INK 1,
1
1640 PAPER 1:FEN 0:CLS
1650 WINDOW #1,3,40,2,9:PAPER #1,0:
PEN #1,1:CLS #1
1660 WINDOW #2,43,79,10,23:PAPER #2
,0:PEN #2,1:CLS #2
1670 WINDOW #3,43,79,2,9:CLS #3
1680 ORIGIN 16,32,16,312,32,224:CLG
(0)
1690 FOR I=3 TO 10
1700 LOCATE 2,1:PRINT CHR$(207)
1710 LOCATE 42,1:PRINT CHR$(207)
1720 NEXT
1730 FOR I=2 TO 39
1740 LOCATE 1,10:PRINT CHR$(207)
1750 LOCATE 1,24:PRINT CHR$(207)
1760 NEXT
1770 FOR I=13 TO 24
1780 LOCATE 2,1:PRINT CHR$(207)
1790 LOCATE 42,1:PRINT CHR$(207)
1800 NEXT
1810 FOR I=42 TO 78
1820 LOCATE 1,24:PRINT CHR$(207)
1830 NEXT
1840 LOCATE 42,12:PRINT CHR$(207)
1850 LOCATE 42,11:PRINT CHR$(207)
1860 INK 1,26
1870 KEY 139,CHR$(13)+"run"+CHR$(13
)
1875 SYMBOL AFTER 199
1880 SYMBOL 200,216,230,246,262,266
,198,198,0
1890 RETURN

```



Para que tus dedos
 no realicen el trabajo duro, M.H. AMS
 TRAD lo hace por ti. Todos los listados que incluyen
 este logotipo se encuentran a tu disposición en un con-
 tento mensual, solicitados.

Si eres lector habitual de esta revista

¡llámame!
de 7 a 10 de la tarde



Te estoy esperando. Tengo muchas cosas que contarte... y muy interesantes. De momento, te propongo la posibilidad de **AHORRAR** más de 1.000 ptas. y, además, con un poco de suerte, **GANAR UNA VESPINO** ¿Qué te parece? Pues esto es sólo un avance. Cuando me llames te contaré más cosas que seguro te gustarán. Pero no te demores, porque a una mujer nunca se le hace esperar. Tienes de plazo hasta el 31 de marzo. Después, habrás perdido tu oportunidad.

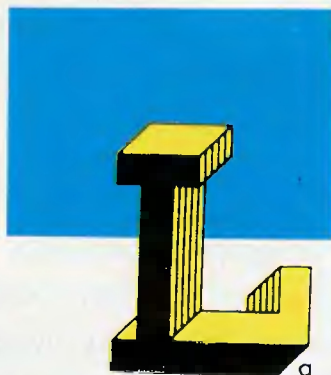


HOBBY PRESS, S.A.

(91) 654 32 11

INSTRUCCIONES DE USO GENERAL Y CONTROL DE LA CPU (I)

Una vez vistas en los últimos capítulos las instrucciones de cambio transferencia y búsqueda, pasaremos a estudiar en el presente capítulo las instrucciones de uso general. Estas actúan generalmente sobre el acumulador y el registro F, produciendo diversos efectos. Debemos decir que el registro F contiene los indicadores y flags.



a primera de estas instrucciones que estudiaremos se representa como sigue:

CPL

Tras la ejecución se invierte el contenido del acumulador, es decir, se produce un complemento a uno del contenido del acumulador.

Para entendernos mejor podemos decir que una vez ejecutado CPL, todos los bits que estaban a uno se ponen a cero y los que inicialmente estaban a cero, se ponen a uno.

Suponiendo que el contenido del acumulador inicialmente es el siguiente:

1 0 0 1 1 1 0 0

tras la ejecución de la anterior instrucción, el acumulador contendrá:

0 1 1 0 0 0 1 1

Por lo tanto el efecto producido se podría explicar también mediante la siguiente expresión:

255-A

donde A sería el contenido inicial del acumulador.

Es decir que el contenido final del acumulador sería igual al resultado

de la diferencia entre 255 y el contenido inicial del registro A.

En el programa número uno podemos ver un ejemplo de actuación. Así pues, colocamos un valor en el acumulador y ejecutamos CPL, con lo que obtendremos en el registro A el complemento a 1 del acumulador.

Inversión de gráficos con CPL

Si observamos con atención el efecto que produce dicha instrucción, y tenemos en cuenta la forma en que podemos obtener gráficos definidos, enseguida nos percataremos de que efectuando CPL a cada uno de los valores que componen el gráfico obtendremos el mismo gráfico pero en modo invertido.

Gráfico normal	Gráfico después de CPL
00011000	11100111
00011000	11100111
00011000	11100111
11111111	00000000
11111111	00000000
00011000	11100111
00011000	11100111
00011000	11100111

El programa número dos muestra un ejemplo de cómo obtener un carácter en su forma invertida

F. L. Frontán





En primer lugar cargaremos en el acumulador el código ASCII del carácter que queremos imprimir, a continuación mediante una llamada al firmware buscaremos la matriz de ese carácter que se encuentra en la ROM inferior, por lo que deberemos habilitarla mediante la llamada a la dirección #B906.

A continuación cogeremos cada uno de los valores que forman la matriz de dicho carácter y efectuaremos sobre ellos CPL, y los almacenaremos en una nueva dirección a la que hemos llamado matriz.

Una vez almacenados los nuevos valores del carácter, pero esta vez en su forma invertida, volveremos a desabilitar la ROM inferior.

Ahora sólo queda colocar esos nuevos valores en la matriz del carácter que deseamos, en este caso el carácter cuyo código ASCII es el 255. Para ello llamaremos a la dirección del firmware que produce este efecto.

Por último sólo nos queda pintar en pantalla este nuevo carácter llamando a la dirección #BB5A.

NEG realiza el complemento a 2

La siguiente instrucción que estudiaremos se representa de la forma siguiente:

NEG

Su ejecución produce el complemento a dos del contenido del acumulador. Esto es lo mismo que restar a cero el contenido del registro A.

Supongamos que el contenido del acumulador en este momento es el siguiente:

1 0 0 1 1 0 0 0

después de la ejecución de la instrucción NEG el contenido del acumulador será:

0 1 1 0 1 0 0 0

Como vemos el efecto producido se podría explicar mediante la siguiente expresión:

0 - A

donde A representa el contenido previo del acumulador.

Un ejemplo práctico de esta instrucción lo podemos observar en el programa número tres. En este caso cargamos el acumulador con el valor 35 y a continuación ejecutamos NEG, y cargamos el contenido actual del acumulador en la dirección de memoria #7000.

Código MAQUINA

Si somos curiosos podemos comprobar cuál es el valor final del acumulador efectuando desde Basic lo siguiente:

PRINT PEEK (&7000)

el valor que obtendremos será 221, es decir el mismo valor que obtendríamos si efectuáramos la operación:

256 - valor del acumulador

Quizá alguien se pueda preguntar el porqué antes hemos dicho que el efecto que producía NEG era el mismo que restar el contenido del acumulador del valor cero, y ahora decimos algo que aparentemente muy diferente.

Pues bien, verdaderamente es diferente en apariencia pero no lo es en la realidad, ya que como sabemos el mayor valor que puede contener un registro simple es de 255, por lo que si incrementamos este registro en 1, en lugar de obtener 256 lo que obtendremos será el valor 0.

Manipulación del Carry flag

Otra de las instrucciones que conforman este bloque es la siguiente:

CCF

Su ejecución produce una inversión del flag Carry del registro F.

Por lo tanto si el flag Carry está a cero antes de la ejecución de CCF, después de ella se pondrá a uno. Asimismo si antes de la ejecución está a uno, después de ella se pondrá a cero.

Observaremos que efectuando dos veces la misma instrucción el contenido del Carry es el mismo que tenía originalmente.

El programa número cuatro muestra un ejemplo práctico de actuación de la anterior instrucción.

En primer lugar miramos el flag Carry antes de que se ejecute ninguna instrucción, si está a uno se imprimirá en pantalla un uno, y si está a cero, imprimiremos ese valor en pantalla. A continuación ejecutamos CCF y volvemos a observar el contenido de dicho flag e imprimiremos en pantalla su valor igual que hemos hecho anteriormente.

Esta instrucción se puede utilizar al inicio de algunas rutinas, en las que se sabe que dicho flag entra con el valor uno, y deseamos que éste esté a cero para que dicha rutina actúe normalmente.

Por ejemplo al llamar desde un programa principal a una rutina de la siguiente forma:

CALL C, rutina

sabemos que al entrar en dicha rutina el flag Carry está a uno entonces efectuando CCF, podemos colocar dicho flag a cero para que la rutina funcione adecuadamente.

La siguiente instrucción que estudiaremos se representa de la forma indicada a continuación:

SCF

Tras su ejecución se pone a uno el flag Carry del registro F independientemente del valor anterior de ese flag.

El programa número cinco nos muestra en pantalla el contenido del flag C del registro F. En primer lugar nos imprime en pantalla dicho valor antes de efectuar ninguna operación, a continuación efectuamos SCF, y volvemos a imprimir dicho valor en pantalla. Naturalmente después de la ejecución de esta instrucción, dicho flag estará a uno.

Es muy útil en rutinas en las que se necesita que el flag C esté puesto a uno. Por lo tanto antes de entrar en rutinas de este tipo efectuaremos SCF.

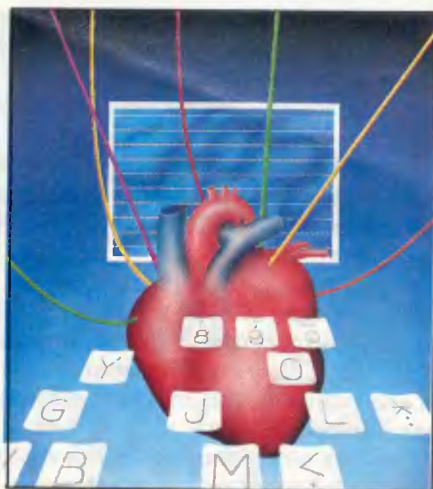
En ocasiones también es factible utilizar las dos últimas instrucciones conjuntamente para producir el efecto deseado.

Por ejemplo en alguna rutina en que necesitamos que el flag Carry del registro F esté puesto a cero y no conocemos el contenido actual de dicho flag. Entonces deberemos hacer lo siguiente:

SCF
CCF

con lo cual obtenemos en primer lugar que dicho flag se ponga a uno y a continuación complementamos su contenido con lo que conseguimos que el flag C del registro F se ponga a cero.

Esto último podemos utilizarlo en rutinas en que se utilicen instrucciones de resta con Carry, en las cuales se efectúa una operación en la cual se resta también el flag C del registro F.



El último programa que hemos preparado muestra un ejemplo práctico de este último caso.

En este caso queremos efectuar una resta entre los contenidos de los registros dobles HL y DE, para ello cargamos dichos registros con los valores deseados.

Una vez hecho esto, queremos que en la resta no influya el flag Carry, por lo tanto, éste debe estar a cero, y para conseguirlo, ejecutamos en primer lugar SCF y a continuación CCF con lo que conseguiremos que dicho flag se ponga a cero, seguidamente podemos efectuar la operación deseada.

PROGRAMAS

```
10 ;PROGRAMA-1
20 ;
A000 3E57 40 ORG #A000
A001 2F 50 LD A,B7
A002 C9 60 CPL
A003  60 RET
```

```
10 ;PROGRAMA-2
20 ;
A000 3E58 40 ORG #A000
A001 CD58B 50 LD A,"X"
A002 CD58B 60 CALL #B8A5
A003 CD06B9 60 CALL #B906
A004 EB 70 EX DE,HL
A005 CD1DA0 80 CALL AQU1
A006 CD07B9 90 CALL #B909
A007 3EFF 100 LD A,FFF
A008 212AA0 110 LD HL,MATR
A009 CDAB8B 120 CALL #B8A5
A010 3EFF 130 LD A,FFF
A011 CD5AB8 140 CALL #B85A
A012 C9 150 RET
A013 212AA0 160 AQU1: LD HL,MATR
A014 0608 170 LD B,B
A022 1A 180 BUC: LD A,(DE)
A023 2F 190 CPL
A024 77 200 LD (HL),A
A025 23 210 INC HL
A026 13 220 INC DE
A027 1BF9 230 DJNZ BUC
A028 C9 240 RET
A02A 250 MATR: DEFS B
```

ETIQUETAS

AQU1 A01D BUC A022 MATR A02A

```
10 ;PROGRAMA-3
20 ;
A000 3E23 40 ORG #A000
A001 ED44 50 LD A,35
A002 320070 60 LD #7000,A
A003 C9 70 RET
```

```
10 ;PROGRAMA-4
20 ;
A000 DC0EA0 40 ORG #A000
A001 D414A0 50 CALL C,UNO
A002 3F 60 CALL NC,CERO
A003 3F 60 CCF
A004 DC0EA0 70 CALL C,UNO
A005 D414A0 80 CALL NC,CERO
A006 C9 90 RET
A007 3E31 100 UNO: LD A,"1"
A008 CD5AB8 110 CALL #B85A
A009 C9 120 RET
A010 3E30 130 CERO: LD A,"0"
A011 CD5AB8 140 CALL #B85A
A012 C9 150 RET
```

ETIQUETAS

CERO A014 UNO A00E

```
10 ;PROGRAMA-5
20 ;
A000 DC0EA0 40 ORG #A000
A001 D414A0 50 CALL C,UNO
A002 37 60 CALL NC,CERO
A003 37 60 SCF
A004 DC0EA0 70 CALL C,UNO
A005 D414A0 80 CALL NC,CERO
A006 C9 90 RET
A007 3E31 100 UNO: LD A,"1"
A008 CD5AB8 110 CALL #B85A
A009 C9 120 RET
A010 3E30 130 CERO: LD A,"0"
A011 CD5AB8 140 CALL #B85A
A012 C9 150 RET
```

ETIQUETAS

CERO A014 UNO A00E

```
10 ;PROGRAMA-6
20 ;
A000 37 40 ORG #A000
A001 3F 50 SCF
A002 21E003 60 CCF
A003 11B403 70 LD HL,1000
A004 ED52 80 LD DE,900
A005 C9 90 SBC HL,DE
A006  90 RET
```

```
10 REM * PROGRAMA 1 *
20 FOR N=&A000 TO &A004
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>180 THEN PRINT "ERROR
EN DATAS"
70 DATA 62,87,47,201,0,0,0
```

```
10 REM * PROGRAMA 2 *
20 FOR N=&A000 TO &A0032
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>1454 THEN PRINT "ERROR
EN DATAS"
70 DATA 62,88,205,165,187,205,6
80 DATA 185,235,205,29,160,205,9
90 DATA 185,62,255,33,42,160,205
100 DATA 160,187,62,255,205,90,187
110 DATA 201,33,42,160,6,8,26
120 DATA 47,119,35,19,16,249,201
130 DATA 0,0,0,0,0,0,0
140 DATA 0,0,0,0,0,0,0
```

```
10 REM * PROGRAMA 3 *
20 FOR N=&A000 TO &A000E
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>2FD THEN PRINT "ERROR
EN DATAS"
70 DATA 62,35,237,68,50,0,112
80 DATA 201,0,0,0,0,0,0
```

```
10 REM * PROGRAMA 4 *
20 FOR N=&A000 TO &A01A
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>05F THEN PRINT "ERROR
EN DATAS"
70 DATA 220,14,160,212,20,160,63
80 DATA 220,14,160,212,20,160,201
90 DATA 62,49,205,90,187,201,62
100 DATA 48,205,90,187,201,0,0
```

```
10 REM * PROGRAMA 5 *
20 FOR N=&A000 TO &A01A
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>05F THEN PRINT "ERROR
EN DATAS"
70 DATA 220,14,160,212,20,160,55
80 DATA 220,14,160,212,20,160,201
90 DATA 62,49,205,90,187,201,62
100 DATA 48,205,90,187,201,0,0
```

```
10 REM * PROGRAMA 6 *
20 FOR N=&A000 TO &A00B
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>422 THEN PRINT "ERROR
EN DATAS"
70 DATA 55,63,33,232,3,17,132
80 DATA 3,237,82,201,0,0,0
```


GRATIS LAS TAPAS AL REALIZAR TU SUSCRIPCION

Ahora, al realizar tu suscripción, MICROHOBBY AMSTRAD te regala estas prácticas tapas especialmente diseñadas para contener tu revista favorita. • NO es necesario recurrir a ningún tipo de encuadernación ni manipulado.

• EN cualquier momento puedes separar un ejemplar determinado y volverlo a colocar en sólo unos segundos.

• SON prácticas y económicas...
... y con diseño especial para satisfacer a la gente inquieta.

(Oferta válida hasta el 31 de mayo de 1986)



Recorta o copia este cupón y envíalo a Hobby Press. Apartado de Correos 232. Alcobendas (Madrid).

Nombre _____ Apellidos _____ C. Postal _____
Dirección _____
Localidad _____ Provincia _____
Teléfono _____ Profesión _____

☐ Deseo suscribirme a **MICROHOBBY AMSTRAD** por un año (50 números) al precio de 6.400 ptas. (IVA incluido). Esta suscripción me da derecho a recibir, **totalmente gratis, las tapas** para contener la obra, valoradas en 670 ptas. (oferta válida hasta el 31 de mayo de 1986).

☐ Deseo recibir en mi domicilio las tapas de **MICROHOBBY AMSTRAD**, al precio de 670 ptas. (IVA incluido).

FORMA DE PAGO. MARCA CON UNA X LA OPCION QUE DESEES.

- ☐ Contra reembolso
- ☐ Mediante tarjeta de crédito VISA. Número de la tarjeta _____ Fecha caducidad de la tarjeta _____
- ☐ Mediante talón bancario o nombre de Hobby Press, S. A.
- ☐ Mediante giro postal n.º _____
- ☐ Mediante domiciliación bancaria
- Banco _____ Sucursal y Localidad _____

Fecha y firma

N.º de cuenta _____

Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «**aformenten**» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

SIN DUDA ALGUNA está abierta a todos.

MODOS DE PANTALLA

A mí me gusta alternar con el tamaño de las letras, por lo que pensaba que en un programa podría alternar con los diferentes MODES, pero no es así ya que un MODE, anula a otro.

¿Es posible usar los tres MODES en un mismo programa?

Si no es así ¿se os ocurre alguna idea para diseñar nuevos modelos de letras?

Alfonso Rincón (Navarra)

Por desgracia, no se pueden usar los 3 modos de pantalla simultáneamente en un programa.

El método de diseñar nuevas letras es sencillo, si tienes en cuenta que un solo símbolo puede constar de varios caracteres, es decir, una «a» puede ocupar cuatro símbolos de los que se definen mediante SYMBOL AFTER.

COMO SALVAR BASIC COMO BYTES

La consulta que quiero hacerles es la siguiente: Cómo se puede grabar un programa escrito en Basic, con el comando Save "", B, dirección inicial, longitud, para luego cargarlo con RUN, ya que lo he intentado varias veces, pero no me ha funcionado.

Ricardo A. Lahosa Laguna (Alicante)

Sí que se puede, pero debes saber dos cosas: dirección de comienzo y longitud.

Un programa Basic comienza en la dirección 170 hexadecimal, y la longitud se almacena en dos posiciones de memoria, QUE DEPENDEN del ordenador.

Para el 464 son: &AE83 y &AE84.

Para el 664 y 128: &AE66 y AE67.

Debes dar la orden así:

464: SAVE «NOMBRE»,B,&170, PEEK(&AE84)*256+PEEK(&AE83)

664: SAVE «NOMBRE»,B,&170, PEEK(&AE67)*256+PEEK(&AE66)

DE DISCO A CINTA

Poseo un 664, y quisiera saber cómo sería posible trasvasar todo el contenido de un disco (programas, archivos, etc.) a cinta, sin tener que ir haciéndolo uno por uno, tanto a través de Basic (con «ltape, save», etc.), como con CPM (usando «file-copy, csave» etc.).

José Prieto Todosantos (La Coruña)

1) Necesitas recurrir a programas comerciales que incluyan esa utilidad, como el TRANSMAT y el PYRADEV. Lamentablemente, de momento no están disponibles en España.

En cuanto a hacerlo tú, necesitarías recurrir al lenguaje máquina, y no tendría la garantía de que lo que hicieras funcionara con todos los programas, especialmente los protegidos.

2) Debes hacerlo de la siguiente manera: salva en el disco la segunda parte del programa, que posteriormente juntarás a la primera como ASCII, mediante la orden SAVE «NOMBRE», A. Previamente, la habrás renumerado con números de línea muy altos, para que no choque con la primera parte.

Ahora, mete en memoria la primera parte y con MERGE la segunda que grabaste en el disco como ASCII.

Una vez hecho esto, renumera el programa y ya está.

INTERRUPCIONES

Quisiera haceros unas preguntas sobre mi CPC 664:

1) Cómo se utilizan y para qué sirven las instrucciones BASIC: WAIT y EI y DI.

2) ¿Cómo se podrían hacer SPRITES desde el BASIC?

3) ¿Cómo se pueden utilizar dos listas de DATAS independientes SIMULTANEAMENTE (por ejemplo para hacer sonar 2 melodías distintas a la vez en diferentes canales)?

Adrián Julia Lundgren (Barcelona)

1) La instrucción WAIT detiene el ordenador un período de tiempo especificado.

El habilita interrupciones, permitiendo que el teclado del Amstrad sea examinado por el sistema operativo, entre otras cosas.

DI las deshabilita.

2) Los sprites no se pueden crear desde el Basic, pero pueden simularse adecuadamente si tenemos pocos. Te remitimos al número 11 de la revista, páginas 15-17.

3) Mediante los comandos EVERY y tal vez AFTER, usando temporizadores distintos.

PUBLICIDAD

New Line
GABINETE DE INFORMATICA

- **Clases de Informática sobre AMSTRAD**
Exclusivamente individuales.
- **Ordenadores AMSTRAD y periféricos**
Los mejores precios
- **Software a la medida**

ZURBANO, 4 ☎ 410 47 63
28010 MADRID

ESCUELA de INFORMATICA APLICADA

"Mister Chip"

CENTRO HOMOLOGADO Y COLABORADOR DEL INEM

- **CURSO de INICIACION**

(6 meses)

Diploma: PROGRAMADOR BASIC-1

- **INFORMATICA BASICA**

(96 horas)

Diploma: PROGRAMADOR EN BASIC

- **PROGRAMACION AVANZADA**

(110 horas)

Diploma: MASTER EN PROGRAMACION

Dirigido a mayores de 12 años.

CIUDAD de los PERIODISTAS. Avda. Herrera Oria, 171 bajo
Frente al Instituto N. Herrera Oria. Tels.: 201 64 09 - 201 93 85

TODAS LAS CLASES SON PRACTICAS
CON ORDENADORES AMSTRAD O SPECTRUM

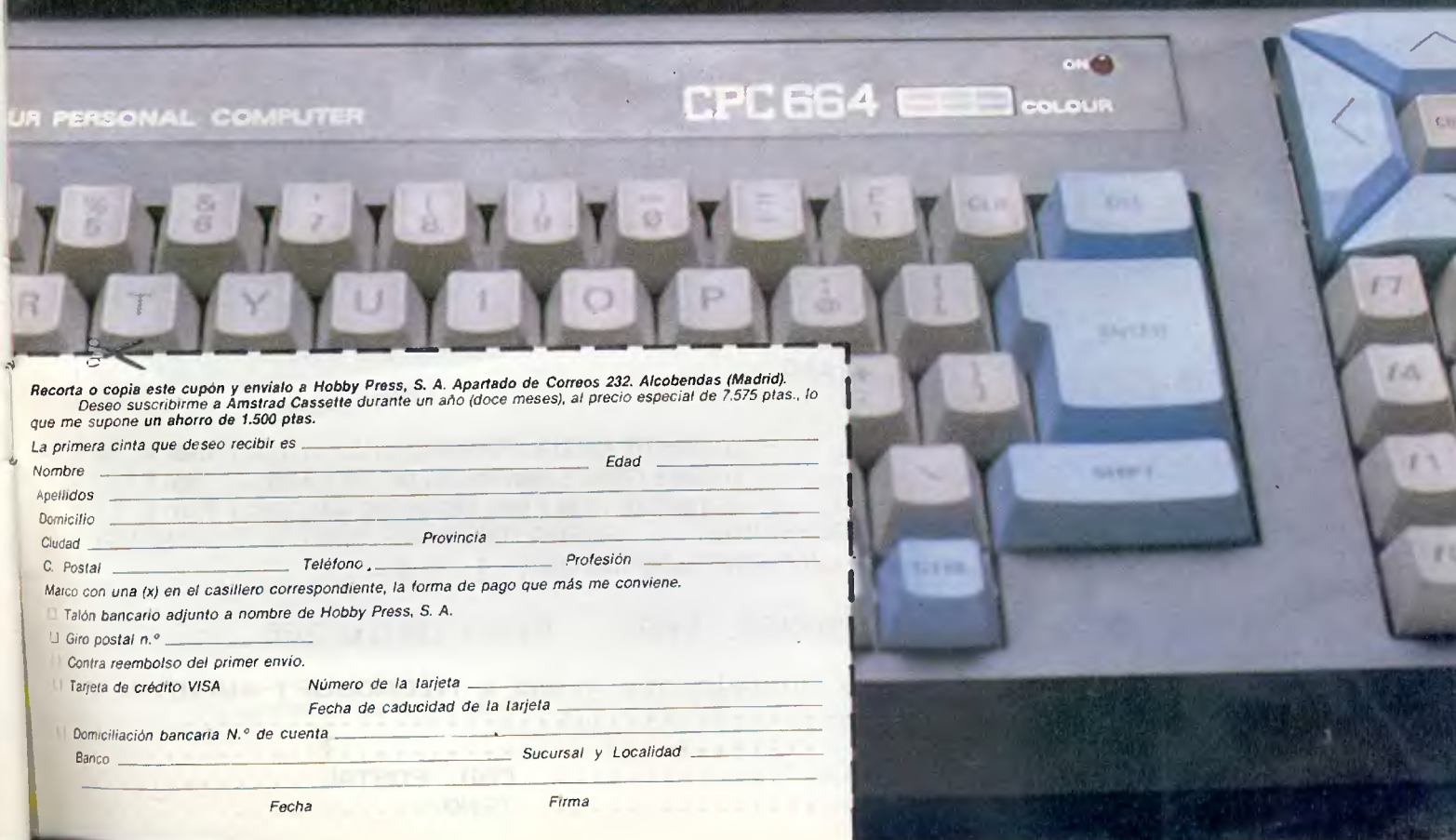
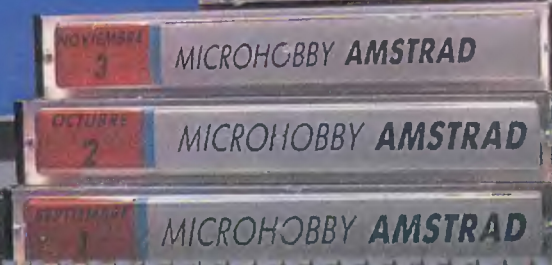
MICROHOBBY AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

- Te ofrecemos un importante ahorro al realizar tu suscripción a Amstrad Cassette.
- Los mejores programas al mejor precio, totalmente desprotegidos con el objeto de facilitar su copia en disco.
- Cada cinta lleva grabados los programas publicados por Microhobby Amstrad durante cuatro números consecutivos (1 al 4, 5 al 8, 9 al 12, etc.)
- Compatibles con los modelos CPC-464, CPC-664 y CPC-6128.

¡AHORRA 1.500 PTAS.!



Recorta o copia este cupón y envíalo a Hobby Press, S. A. Apartado de Correos 232. Alcobendas (Madrid).
Deseo suscribirme a Amstrad Cassette durante un año (doce meses), al precio especial de 7.575 ptas., lo que me supone un ahorro de 1.500 ptas.

La primera cinta que deseo recibir es _____ Edad _____

Nombre _____

Apellidos _____

Domicilio _____

Ciudad _____ Provincia _____

C. Postal _____ Teléfono _____ Profesión _____

Marco con una (x) en el casillero correspondiente, la forma de pago que más me conviene.

☐ Talón bancario adjunto a nombre de Hobby Press, S. A.

☐ Giro postal n.º _____

☐ Contra reembolso del primer envío.

☐ Tarjeta de crédito VISA Número de la tarjeta _____

Fecha de caducidad de la tarjeta _____

☐ Domiciliación bancaria N.º de cuenta _____

Banco _____ Sucursal y Localidad _____

Fecha

Firma

Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, **MERCADO COMUN** te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En **MERCADO COMUN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a:

HOBBY PRESS, S.A.

AMSTRAD SEMANAL.

Apartado de correos 54.062

28080 MADRID

¡ABSTENERSE PIRATAS!

Vendo el siguiente lote de programas al precio de 2.750 pts.: Decathlon, Easi-amsword (Procesador de textos), Fred, roland in then caves The Galactic plage, Harrier attack. Llamar al Tel. (976) 34 61 95. O escribir a Joaquín López Vela. C/ Parque de Roma, F-1 - 8-G. 50010 Zaragoza.

Desearía cambiar el DALEY THOMPSONS DECATHLON (original), por uno de estos programas:

Exploding fist.
Rocki o Frank Bruno's Boxing.
Beach Head.
Tenis.
Knight Lore.
Pole Position.
Raid over Moscow.
Dragontore.
Southern Belle.

También me interesarían algunos otros títulos. Interesados llamar a: José María Crespo. C/ Jaime Fabrè, 2 - 10.º-1.ª. 08019 Barcelona. Tel. 305 69 28.

Vendo Amstrad CPC-664 con monitor de fósforo verde. Nuevo. Con garantía oficial. Regalo Pascal en disco. Por 89.000 ptas. Tel. (960) 285 17 84 de Oliva (Valencia). Preguntar por Pedro. Transportaría a cualquier punto de la provincia de Valencia.

Vendo Amstrad 464, monitor fósforo. En garantía. Regalo juegos y manuales de uso y Basic Amstrad, 55.000 ptas. Eduardo Ruiz de Velasco. Tel. (91) 204 52 76.

Vendo Amstrad 464-F verde con modulador MP-1 para televisión color y B. & N., 24 juegos

MICROSOFT-HARD, S.L.

Apartado 23.406, 08080-Barcelona, Tfn. (93) 348 04 07 (Tardes de 5 a 9)

MANTENGA SU MICRO COMO NUEVO CON UNA DE ESTAS PRACTICAS FUNDAS

- ☐ SPECTRUM 16/48..... 330 ¢
- ☐ SPECTRUM PLUS..... 460 ¢
- ☐ SPECTRUM 128 K..... 855 ¢
- ☐ SINCLAIR Q.L. 1.180 ¢
- ☐ TECLADO SAGA 1 570 ¢
- ☐ TECLADO SAGA 3 715 ¢
- ☐ COMMODORE y VIC 20... 665 ¢
- ☐ HIT-BIT 665 ¢
- ☐ SEIKOSHA SP 800/1000: 900 ¢
- ☐ IMP. AMSTRAD DMP-1... 1.205 ¢
- ☐ IMP. RITEMAN F+/C+... 825 ¢
- ☐ IMP. RITEMAN 10/120.: 785 ¢
- ☐ AMSTRAD CPC 464 (V) ☐ (C) O... 2.262 ¢
- ☐ AMSTRAD CPC 664 (V) ☐ (C) O... 2.262 ¢
- ☐ AMSTRAD CPC 6128 (V) ☐ (C) O... 2.262 ¢
- ☐ AMSTRAD PCW 8256 3.250 ¢

(MARQUE CON UNA "X" LAS OPCIONES DESEADAS)

Gastos de envío: 150 ¢

☐ CASSETTE ESPECIAL ORDENADOR..... 4.650 ¢

☐ CABLE CASSETTE PARA AMSTRAD CPC 664 y 6128... 850 ¢

☐ CASSETTE Y CABLE PARA AMSTRAD CPC 664 y 6128.: 5.400 ¢

FORMA DE PAGO CONTRA REEMBOLSO

TODOS ESTOS PRECIOS LLEVAN INCLUIDO EL I.V.A.

INTERESANTES CONDICIONES PARA DISTRIBUIDORES

Recorte o copie este anuncio y envíelo hoy mismo a **MICROSOFT-HARD, S.L.**

-NOMBRE Y APELLIDOS

-DOMICILIO

-LOCALIDAD

-PROVINCIA

COD. POSTAL

TFNO.

MICRO-1

C/ Duque de Sesto, 50. 28009 Madrid
Tel.: (91) 275 96 16/274 53 80
(Metro O'Donell o Goya)

el IVA lo paga
MICRO-1

SOFTWARE: por cada programa GRATIS ¡¡1 BOLIGRAFO CON RELOJ DE CUARZO!!

HYPER SPORTS	2.300 ptas.
TORNADO LOW LEVEL	1.950 ptas.
EXPLODING FISTT	2.300 ptas.
JUMP JET	2.495 ptas.
ZORRO	2.600 ptas.
SABREWULF	1.650 ptas.
GHOSTBUSTERS	1.950 ptas.
GYROSCOPE	2.300 ptas.
HIGHWAY ENCOUNTER	1.750 ptas.
HIGHWAY ENCOUNTER DISCO	3.300 ptas.

DYNAMITE DAN	2.100 ptas.
RAID OVER MOSCOW	2.300 ptas.
THEY SOLD A MILLION	2.500 ptas.
FIGHTER PILOT	1.975 ptas.
MASTER OF T. LAMP	1.950 ptas.
NIGHTSHADE	1.950 ptas.
HACKER	1.950 ptas.
SUPER TEST	2.300 ptas.
MAPGAME	2.700 ptas.
TONADO LOW LEVEL DISCO	3.300 ptas.

JOYSTICK QUICK SHOTT II... 2.295 ptas.
JOYSTICK QUICK SHOT V ... 2.595 ptas.

PC-COMPATIBLE IBM 256 K
MONITOR FOSFORO VERDE
2 BOCAS DISKETTE 360 K
SOLO ¡¡243.900!!

TAPA METACRILATO PARA
TECLADO ¡¡1.900 ptas.!!

UNIDAD DISKETTE 5.25"
¡¡45.900 ptas.!!
(incluido controlador)

LAPIZ OPTICO
¡¡4.900 ptas.!!

IMPRESORA MARGARITA
¡¡49.900 ptas.!!

CASSETTE ESPECIAL
ORDENADOR 5.295 ptas.

PRECIOS SUPER-EXCEPCIONALES PARA
AMSTRAD CPC-472 Y CPC-6128
¡¡LLAMANOS, TE ASOMBRARAS!!

IMPRESORAS
¡¡20% DTO. SOBRE P.V.P.!!

SINTETIZADOR DE VOZ
Y AMPLIFICADOR:
7.900 ptas.

MODULADOR TV
8.400 ptas.

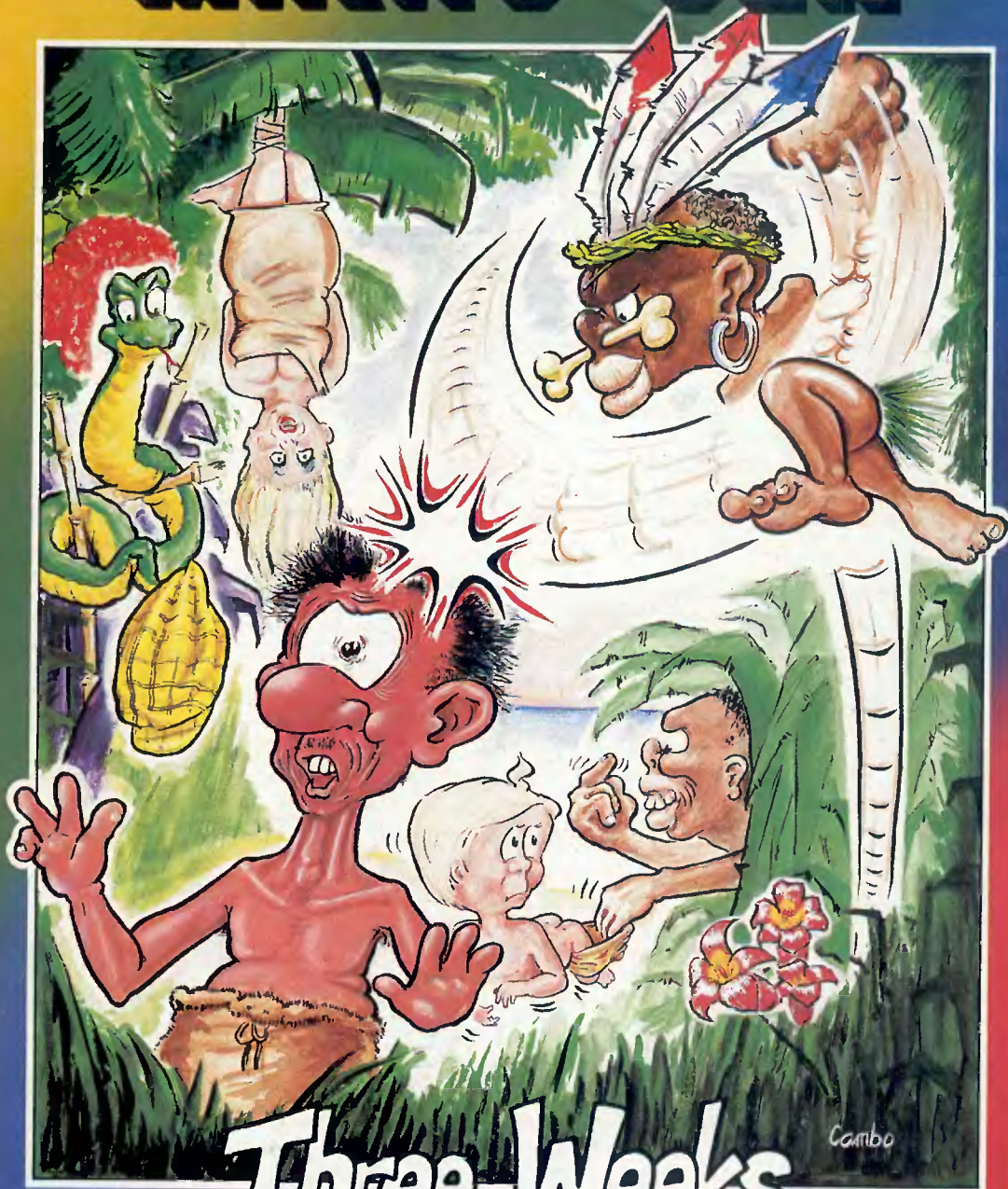
CINTA C-15 ESPECIAL
ORDENADOR 85 ptas.
DISKETTE 3" 990 ptas.

UNIDAD DE DISCO 3" CON
CONTROLADOR: 49.900 ptas.

Libros:	
Curso autodidáctico Basic I	2.525 ptas.
Curso autodidáctico Basic II	2.525 ptas.
Programando con Amstrad	2.195 ptas.
Juegos sensacionales Amstrad	1.950 ptas.
Hacia la Inteligencia Artific.	1.295 ptas.
Música y sonidos con Amstrad	995 ptas.

SI BUSCAS LO MEJOR **ERBE** Software LO TIENE

MIKRO-GEN



Three Weeks in Paradise

**LA FAMILIA WALLY ATACA DE NUEVO CON LA MEJOR
Y MAS DIVERTIDA AVENTURA QUE PUEDES IMAGINARTE**

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE, SANTA ENGRACIA, 17. Tel: 447 34 10. DELEGACION BARCELONA, Avd. MISTRAL, 10. Tel. (93) 432 07 31